Agent-Based Modelling of Public Space Activity in Real-Time

Kostas Cheliotis

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Centre for Advanced Spatial Analysis

University College London

July 16, 2018

I, Kostas Cheliotis, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

Understanding how urban space is used by its inhabitants is vital in improving the overall quality of a city's built environment, as it can highlight needs and requirements of everyday life to be addressed in any urban development. Our investigation of urban activity is often approached through spatial models and simulations on the one hand, and urban data on the other. The work presented here explores potential combinations of the two, by coupling urban models with real-time urban data feeds for continuous short-term forecasting of urban activity.

This aim is approached through the development of a model of activity in urban public spaces using the agent-based modelling paradigm, calibrated to real-time data input, and applied to the simulation of current activity in public spaces at a fine spatio-temporal scale. Observations about human spatial behaviour are identified in the literature on public spaces and implemented within a 3D modelling framework, thereby extending existing pedestrian and crowd agent-based modelling approaches. Furthermore, a review and evaluation of real-time data feeds pertaining to activity in public spaces is performed, focussing on open and publicly available datasets, and a forecasting model is developed using social media and other datasets as a proxy for current user activity. The resulting real-time model of public space activity is then evaluated through two case studies focussing on two major urban parks in London, UK.

The model performs well in capturing park visitor activity in terms of spatial dispersion. Real-time data feeds examined are found to be capable of capturing park

visitor activity to some degree; however they are found to be inadequate in supporting a fully fledged, detailed real-time model of public space activity.

Finally, potential future trajectories of the approaches are identified in the increasing availability of online 3D mapping data when combined with advances in computational efficiency and data availability, in extending current data visualisation approaches into expansive, fine-scale simulations of real-time urban activity.

Impact Statement

This thesis presents a model for simulating human activity in public spaces in real-time. In doing so it addresses and reviews multiple fields, and therefore identifies potential impact in multiple instances, both within and outside academia.

In academic context, this thesis reviewed existing literature and produced reviews of two fields: First, it reviewed findings on human activity and interaction in public spaces, as presented in multiple studies. It produced a summary of said findings, covering aspects of human navigation and movement in open space, grouping and crowding behaviour in public, human-environment and human-human interaction. Secondly, it reviewed literature on models of pedestrian movement, and produced a summary and classification of agent-based models of pedestrian movement.

Furthermore, this work produced a real-time modelling framework for continuous forecasting at high temporal fidelity. Preliminary work on this model including calibration and initial evaluation was presented at the 10th International AAAI Conference on Web and Social Media (ICWSM)[1] and a version was published at the conference workshop proceedings (Cheliotis, 2016).

In non-academic context, this work identifies two potential applications. First, it presents a tool for visualising park visitor activity in real-time without requiring the installation of additional sensing and monitoring devices, relying instead on publicly available data. Such a tool would be suitable for use in public spaces (such as parks) to monitor visitor conditions for safety and security purposes as well as

[1] http://www.icwsm.org/2016/

measuring park performance and accessibility, with minimal added cost.

Secondly, this work presents a simulation framework that captures user activity in public spaces. Whereas existing models focus mainly on user flows, the model presented here takes into account stationary activities and presents a more holistic model of *public space use* in 3D environments. Such a model would be valuable in the built environment and design professions, as a tool for exploring ”What If?” scenarios and evaluating proposed designs of public and open spaces from the user perspective.

Acknowledgements

I would like to thank my supervisor, Professor Andrew Hudson-Smith for his continuing support and encouragement, whose advice helped guide this work to completion. His insightfulness was invaluable in helping me clarify my thought process at times, his ambition inspired me to look forward, and his patience provided a safe environment to work, experiment, and most importantly make mistakes in. Similar thanks goes out to Professor Mike Batty, my secondary supervisor, whose knowledge and willingness to discuss meant that he was always able to point me to where the answer lies. Thank you to Professor Duncan Wilson as well, my external supervisor and Director at the Intel Collaborative Research Institute in London, which funded this research in part through a UCL Impact Award, and who provided me with the opportunity to be a part of and collaborate with a wide network of researchers.

I would like to thank my colleagues at the Centre for Advanced Spatial Analysis, and particularly my fellow doctoral candidates, for the many engaging discussions that took place and for the opportunity to learn alongside them. Their willingness and constant curiosity to discuss anything and everything was a lesson in itself, and more often than not provided the missing piece of information I did not know I was looking for.

Most of all, thank you to my parents, Angelliki and Dimitris, and my grandmother Aglaia, whose unwavering support, encouragement, and love allowed me to want to push further.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Context

For urban dwellers, a large number of daily activities occur in the urban public space. Starting with the most frequent daily activity, navigation and movement in the city happens almost exclusively in public space, be it cycling, driving, riding public transport, or most importantly walking, as almost all modes make use of streets as places dedicated to common use. In addition to movement, a wide range of additional activities take place in the urban public space, including leisure and recreational activities, cultural activities, information exchange, commercial activities, and social interaction, with a lot of them happening at the same time, often subconsciously, as part of urban life.

Considering then the number of activities and interactions that are part of urban life, it is of interest to study the properties and characteristics of cities which can have an effect on such activities, in order to plan accordingly and provide better conditions for urban dwellers. Given the importance of public spaces as the environment which hosts such activities, it follows then that one of the main requirements of a successful urban public space is (or at least should be) for it to be 'habitable'/usable, or at the very least enable and allow people to 'spend time' in it. One of the ways that space itself can have an effect on the activities taking place in it is through

its physical properties, which in the case of urban public spaces constitute all the boundaries, paths, obstacles, materials, and in general the form of the space itself. These physical properties of space, when materialized in urban public spaces, are addressed through the field of urban design.

Urban design, in its contemporary definition, is a relatively new field which emerged within the last century. Rowley (1994) places the emergence of the term in the late 1950s), through the need to handle the unprecedented urbanization brought as a result of mass industrialization. It grew extensively with the rise in popularity of the Modernist movement and played an important part during the inter-war and post-war periods. Following that period, contemporary urban design received major criticism (Jacobs, 1961) and was seen as a catalyst of many of the problems evident in cities at the time, due to their emphasis on the automobile and the disassociation of streets and public life (Southworth and Ben-Joseph, 2003, Marshall, 2005). In recent years however, there has been a resurgence in the need for good quality urban spaces, as evident in various plans around the world for revitalization of areas in decline, and the emergence of the concept of 'third' places (Banerjee, 2001), even if such spaces are often not public at all (e.g. Privately Owned Public Spaces (POPS)).

Urban design is an interdisciplinary field which stands at the intersection of a range of related fields, including architecture, landscape design, urban planning, and sociology among others, with many ambiguities regarding its scope and focus (Madanipour, 1997). As such, definitions on what urban design is vary depending on the starting point (Marshall, 2012). For the purposes of this work, urban design will be considered here as *the process which produces the form of public spaces at the human scale*, with a specific focus on open spaces. Given its outcome, it plays an important part in the overall planning and shaping of the cities around us, as it has the capacity to facilitate and define interactions both between people and the built environment, and between people themselves.

As stated earlier, urban public spaces host a wide range of different activities. Fur-

thermore, contemporary planning approaches in the UK often aim to include and accommodate a mixture of different activities, for large parts of the day, in an attempt to enhance the vitality of spaces. Where successful, such cases exhibit an interplay between the different people and activities in a space, often requiring a balanced mix between actors and activities. As such, it is hypothesized that successful urban public spaces are able to host a large set of heterogeneous activities.

It is understandable then how the continuous study of public spaces plays an important role in improving the urban environment. However, in addition to the study of the design of spaces, as seen in architectural and urban design theory in the past 50 years, equally important is also the study of the interactions that take place in a space. By studying the product of urban design from a human-centric perspective, we can examine the realized potential of a space, or how a public space is ultimately used by its intended users. This approach highlights the impact a place ultimately has, and can help identify successful urban design approaches and further highlight unforeseen advantages in a particular design. Given however the complexity and apparent randomness often evident in spaces containing human interaction, recording such behaviour has often been best achieved through traditional means, such as direct observational studies and site surveys; as usual in urban studies until recently data collection required a clipboard, clicker counter, and a large team of people. Therefore, a point needs to be made here, that *capturing public space usage is a task which requires substantial technical and manual labour*.

Furthermore, the interconnectedness of activities and the effect different conditions can have on the same space can make it hard to identify causality in studies of space use. In addition to this, the rigidity of urban form does not offer a large degree of experimentation on the part of researchers. As such, it is often hard to study space use under extensive scientific rigour, a fact that is also evident in fields such as ecology, social studies, etc. which focus on dynamic systems in the real-world. This often means that space use studies will form hypotheses on the dynamics of spatial activity, but *are often unable to advance to the next step of testing them in a*

controlled environment.

Thus, two main problems in public space use studies have been identified. First, *extensive data on public space use is difficult to collect.* Second, *the evaluation of hypotheses on the dynamics of spatial activity is difficult or impossible to perform in realistic conditions.* In the following sections, this work will discuss how recent advances in computational modelling, as well as the advent of Big Data and Real-Time urban data, can potentially provide solutions to the two problems presented here.

Models of cities and spatial systems are important to the geographic sciences and urban planning, especially given the rapid urbanization taking place in recent years. A historic review of urban models would be impossible to do in this context, as the field can be traced back to 1933 with Christaller's Central Place Theory, or even earlier, to 1826, with von Thunen's model of agricultural land use, and is outside the scope of this work. Rather, this work will be involved with computational models of urban systems, as they have emerged in the last 60 years or so. During this time, models have been developed which capture a wide range of properties of urban space, from spatial economies, to traffic flows, to environmental aspects, to land use, among others.

It is understandable then even at this point how computational modelling approaches can be applied to the system in question here, which is public space use at the human scale. Aspects of interest in this system include pedestrian flows, densities, use of space and its spatial distribution, visibility, etc, and such aspects could very well be captured and simulated in broad strokes through many of the existing modelling paradigms. Indeed some computational approaches developed within the last 40 years have studied some of the aspects mentioned here, and in fact advanced the field extensively, as can be seen in the concept of the isovist as used in Space Syntax studies to measure visibility, or flow models of pedestrian movement.

However, as stated previously, such approaches often only capture activity in broad strokes, simulating mainly aggregate activity. During its development, the field of computational urban modelling has moved from early macroscopic static models of urban systems, to more recent microscopic disaggregate models focussing on the dynamics of various urban systems at fine scales, a direction which has been enabled in part due to advances in computing power. It is these later disaggregate dynamic models, which have risen to prominence within the last 20 years or so, that this thesis will focus on, as it has been demonstrated that such approaches are much more suitable in studying a dynamic system at high spatial resolution, such as public space use at the human scale.

One such approach which is of interest here is the agent-based modelling paradigm. A short introduction to agent-based modelling will be presented here to highlight its relevence in this work, with a thorough discussion offered in a following chapter (section 3.2). In agent-based models, *"a system is modeled as a collection of autonomous decision-making entities called agents, where each agent individually assesses its situation and makes decisions on the basis of a set of rules"* (Bonabeau, 2002). This microscopic approach introduces stochastic and dynamic behaviour in the modelled system, and provides potential for the inclusion of heterogeneous characteristics among the agents. As such, agent-based models can provide a test bed for scenarios in silico, allowing for the simulation of systems that would otherwise be difficult to examine.

With characteristics as discussed above, agent-based models can potentially be a fitting analytical approach in the study of public space use. First of all, public spaces host a wide range of heterogeneous activities within the same shared area, which agent-based models can incorporate through agent definition. Furthermore, users of public space act according to their own personal preferences, often by adapting to the conditions around them. This stochastic characteristic of public space activity can be captured again in agent-based models through the definition of agent behaviours and interaction rules. Additionally, public spaces are inherently dy-

namic places, with behaviour changing every minute, as a result of internal (eg. crowding) and external (eg. weather, time of day) conditions. Again, such dynamic behaviour is exhibited in agent-based models. Finally, observed overall activity in public spaces is considered to be the result of individual actions and reactions, as no single individual user is actively working towards a predefined state of overall activity. As such, aggregate activity emerges through local behaviour, which is a characteristic agent-based models are designed to capture.

Given the above comparisons, agent-based models can be considered a valid and useful approach to the study of public space use as defined earlier. Furthermore, they can offer an additional advantage to this study: They can provide a platform in silico for conducting experiments, through which hypotheses in public space use studies can be tested, a process which would be difficult if not impossible to do in the real-world. As such, agent-based models can provide a solution to the second problem of public space use studies identified earlier, as the evaluation of hypotheses.

A final note needs to be made here regarding the development of computational models: One important requirement for developing computational models is the availability of extensive datasets and information on the system of interest. Such datasets are needed for the evaluation and calibration of the model, to ensure that a phenomenon is captured and simulated adequately, while maintaining predictive capabilities and applicability to related scenarios (i.e. not overfitting). Therefore, agent-based models and public space use studies share another similarity, in their requirement for detailed datasets.

This requirement for large datasets in both public space use studies and the development of urban models has certainly influenced the extent to which each field can grow. Although by no means inhibitive, the relative scarcity of datasets meant that data collection played a more integral part in the overall study, potentially affecting the direction of the research. Under this light, it is always of interest then to examine

potential opportunities in new data availability. One such opportunity is identified in the wealth of data being made available today, through the rise of big data and the adoption of information and communication technologies by cities (termed Smart Cities) in recent years.

The advent and consequent growth of the semantic web (Tim Berners-Lee et al., 2001) in the past decade and a half has brought about a new paradigm in regard to communications and information exchange. Through the establishment of standards for data formats and communication protocols, it became much more feasible to share and receive information. Furthermore, advances in mobile computing technology introduced powerful computing devices into everyday life in the form of smartphones and handheld devices, which are able to capture, generate, and share in unprecedented volumes of data. Finally, the development and subsequent installation of specialized sensors for the monitoring and managing of large systems introduced networked infrastructure systems, which, when met with advances in microprocessors and networking capacity, enabled the emergence of ubiquitous computing in what is now called the Internet of Things (Gubbi et al., 2013). All these different aspects of capturing and sharing data have seen application in the urban realm in smart city schemes, where data on urban systems is continuously used to enhance everyday life.

Within this cloud of big data then, it is the interest of this work to identify and examine potential datasets which might aid in the study of public space use. Indeed, such opportunities are initially identified in various datasets: Urban transport infrastructure systems provide frequent updates on the state of the transportation network, environmental services provide information on the quality of the urban environment, networking devices capture information on visitor crowds in various places, and people themselves share information with their friends, acquaintances, and the public, over social media networks.

In addition to the volume of such datasets, there is another characteristic that is of notable interest to this work: Data discussed here is often shared at the moment

of capture, with a high degree of temporal resolution. As such, these datasets can provide us with a view into the workings of the world around us at right this instant, i.e. in *real-time*. By examining and analyzing data in real-time, a simulation can be developed to run in real-time itself, i.e. simulating the phenomenon of interest concurrently to the phenomenon taking place. Such a prospect has the potential to be of notable value in urban studies, especially considering the continuous predictive capabilities of such models. It is then this real-time element of big data, along with the disaggregated fidelity it brings, that this work shall focus on, for two reasons: Firstly, *it can provide an indicator of the small-scale system dynamics that are of interest here*. Secondly, *it can help develop real-time simulations*, which would aid in the comprehension and dissemination of the finer workings of urban processes. As such, real-time datasets can offer a solution to the first problem of public space use studies identified earlier, along with the similar Agent-Based Modelling requirement, that of data availability.

1.2 Research Focus

The three main fields this work will focus on have now been introduced. They are broadly defined as follows:

1. **Public Space Use Studies**: The field of study focussing on human interaction and activity in space. This field applies a human-centric/user-centric approach, examining interaction among the different users of a space, and the interaction between users and their environment. Further focus is placed on the spatial configuration of activity and interaction on the one hand, and the effect of the physical form on said activity on the other. The majority of studies take place in urban environments.

2. **Agent-Based Modelling**: A branch of computational modelling, designed to study systems of a stochastic nature. Its defining characteristic is the investigation of aggregate system properties as the result of the interaction between

individual autonomous entities within the system, called agents.

3. **Real-Time Data**: Information that is published/delivered at the moment of capture, in this case relating to information on the urban environment. Recent advances in technological fields have made the capturing and broadcasting of data much more feasible, resulting in the emergence of a host of services which deliver diverse data sets and indicators of urban activity, as the activity takes place (i.e. in real-time). This has resulted in an unprecedented volume of detailed data on various aspects of urban activity.

Furthermore, a number of shortcomings and limitations have been identified in each field. More specifically, data on public space use is often gathered through extensive observational site surveys and as such is difficult to collect, while conducting further experiments is often infeasible, due to the broadness of the field. Agent-Based Modelling, along with other computational approaches, require large datasets, in order to calibrate and evaluate the models.

This research will explore potential connections between the three fields, as multiple instances have been identified where characteristics and findings from one field can enhance and complement parts in the others. The purpose of this work is to bring these three together, with an overall goal to develop a better understanding of how we use our public spaces.

1.3 Aim, Structure, and Objectives

Aim This work will examine connections between agent-based models and real-time urban datasets, applied in the study of activity in public spaces. Within this context, the aim of this work is then to develop an *Agent-Based Modelling* framework of *Public Space Use*, calibrated using *Real-Time Data* streams, and applied to a simulation of current activity and conditions of public spaces; a *Real-Time Simulation of Public Space Activity*.

Structure This research aim will be approached through a 3-part structure linking the 3 main fields of interest of this work. The three parts are as follows:

1. **Codification of human behavioural rules in public spaces** as they have been observed and postulated in relevant literature. These behavioural rules will form the building blocks with which the computational simulation of public space activity will be developed.

2. **Development of a Simulation Framework of Public Space Activity**, through the application of the above codified behavioural rules into an agent-based model. This model will capture public space activity at the individual level, and output overall spatial activity.

3. **Extension of the framework into a Real-Time Simulation of Public Space Activity**, through the application of real-time data streams to the simulation framework. Or to put it differently, calibrating the simulation to run based on data published in real-time. This will result in a simulation which will provide an estimation and visualisation of current activity in a space.

Objectives The overall aim will be pursued through a number of objectives, which are defined as follows:

1. Review existing literature on studies of public space use, and identify prevailing hypotheses of public space user behaviour and rules of interaction.

2. Review spatial modelling approaches, and identify appropriate methodologies for modelling the activity of individuals in public spaces.

3. Review potential real-time data sources pertaining to activity in public spaces, and develop methodologies to capture and analyze selected datasets.

4. Develop a general framework for real-time models of public space activity.

5. Based on the outcomes of objectives 1 & 2, codify identified behaviours, build a spatial model of public space activity, and couple with the general

framework developed in 4.

6. Through the combination of objectives 3 & 4, couple the general framework model developed so far with real-time data feeds.

7. Apply the real-time model of public space activity, and evaluate against real-world conditions.

1.4 Thesis Outline

An overview of the thesis organization is presented here, as laid out across the 11 chapters. The thesis is organized in three parts: *Theory*, *Methods* and *Applications*. The first part, *Theory*, which includes Chapters 2, 3, and 4, begins with a literature review on each of the three fields of interest: Public Space Use (PSU), Agent-Based Models (ABMs), and Real-Time Data (RTD), and establishes the theoretical framework for the rest of this work. The second part, *Methods*, in Chapters 5, 6, and 7, describes the methodologies used to develop *Agent-Based Models of Real-Time Public Space Activity* in this work, and essentially begins to formulate the specific tools born following the theoretical investigation. The final part, *Applications*, which includes Chapters 8, 9, and 10, presents a record of the two case studies undertaken in this work, along with findings and an extended discussion on results, methods, and lessons learned. It constitutes a direct real-world application of the tools presented in the previous part. A short description of each chapter in the thesis is offered here, in order to illustrate how each chapter addresses each of the thesis objectives.

The following chapter, *Chapter 2: Understanding Public Space Use* lays the theoretical groundwork of this work. It further expands on the importance of public space through a review of prominent urban theorists' work, and introduces observational studies as an analytical methodology to the study of urban space. The literature review focuses on studies which placed human behaviour and interactions in public spaces as the main focal point, examined both as a result of design principles,

and as inherent human nature within sociocultural norms. The chapter concludes with a categorization of observations and hypotheses on what drives human activity and behaviour in public spaces, thus fulfilling objective 1.

Chapter 3: Computational Models in Urban Studies provides a short review on computational modelling approaches in urban studies, and a literature review of the agent-based modelling paradigm in particular. In doing so, it examines the applicability of the ABM paradigm to this particular scenario (public space use), thereby completing objective 2.

Having established the area of interest and the technical/analytical tools that will be used in this work, the following chapter (*Chapter 4: On Real-Time Data*) addresses the datasets and Real-Time Data (RTD) sources that will be used. The chapter begins with a clarification section, first by examining the different meanings of the term *'real-time'*, and second by defining the term as it will be used in this work. Following that, RTD is identified in the contemporary context of smart cities and the wider field of big data, and the various aspects of these multi-faceted terms are discussed and untangled via an analysis of their apparent dichotomies. The chapter concludes on the applicability of specific RTD sources to the study of Public Space Use (PSU) through ABM, thus completing the first part of objective 3.

Chapter 5: Real-Time Simulation Methodologies outlines the framework for a real-time disaggregated model of public space activity. This is achieved through a two step process, through a predictive model of aggregate activity, followed by a spatial disaggregation model of individual activity. This chapter provides a framework for a Real-Time Model of Public Space Activity, thus fulfilling objective 4.

Chapter 6: Data Collection and Analysis discusses all aspects relevant to datasets used in this work. It presents all the different data sources, along with methods developed for collecting the data, where applicable. Initial analysis of the datasets is also presented, providing an evaluation of the applicability scope of potential sources. With this chapter, the second part of objective 3 and objective 6 are com-

pleted.

Chapter 7: Modelling Spatial Behaviour focusses on the methodologies employed in the development of spatial behaviour and activity models. It presents the development of the ABM framework that is used in this work, which is achieved by implementing the codified human behavioural rules identified during objective 1 in an ABM context. With this, objective 5 is completed.

The following two chapters, *Chapter 8: Case Study 1 - Hyde Park* and *Chapter 9: Case Study 2 - Queen Elizabeth Olympic Park*, present the application of all methodologies developed earlier to real-world scenarios, and essentially document the development of the two case studies undertaken in this work. Both chapters share a similar (if not yet identical) structure. The real-time model framework is calibrated and adapted to represent activity of the area in question. Following that, simulation output is evaluated against control real-time data. With this, the final objective (7) is fulfilled.

The penultimate chapter, *Chapter 10: Discussion on Case Studies*, offers a discussion on this endeavour. It evaluates the datasets used in both case studies in terms of accessibility, applicability, veracity, etc. Furthermore, an evaluation of the developed framework and models is presented, identifying problematic areas. Finally, a discussion on the results of the two case studies is offered, highlighting interesting points and notes.

The final chapter, *Chapter 11: Conclusion*, presents a summary of the findings and major contributions, and readdresses the statements of the opening chapter with a critical view. It concludes with a discussion of potential future work.

Part I

Theory

Chapter 2

Understanding Public Space Use

This chapter discusses existing literature on the study of public space use. In doing so it establishes the base theoretical framework of this thesis. More specifically, it establishes the relevance of this work in the greater context of planning and designing communal urban spaces that are fit for use. It argues that urban public spaces are ultimately designed and built to be used by the people of a city, and as such all relevant tools should be employed in order to maximize the success of such spaces in terms of end-user need fulfilment. It has been previously established that models and simulations are some of these tools available, and that their application can enhance the place-making potential of planners and designers. With this in mind, it is equally important then to review findings on public space user behaviour as it has been identified through observational and empirical studies, in order to highlight public space user needs. Such data and knowledge will play an important role in the consequent development of models and simulations of public space activity, as it will be used to both inform and optimize, and subsequently verify the models developed.

The first section (*2.1: The Significance of Public Space*) in this chapter establishes the importance of public spaces in urban life as the environment and mediator through which most of urban life takes place. Furthermore, prominent aspects and characteristics of public space are presented, as discussed by urban theorists, and

the scope within which this thesis will approach public space is defined. The following section (*2.2: Studying Human Behaviour in Public Spaces*) reviews relevant literature regarding human behaviour in public spaces. Literature is divided in three parts, the first addresses the navigation and locomotion of humans in spatial environments, the second discusses stationary and active engagement human activities in public spaces as affected by social and design aspects of spaces, and the third examines the apparent distances observed in social interactions. The third section in this chapter (*2.3: Categorization of Human Behavioural Observations*) presents a summary and codification of all relevant observations on human socio-spatial behaviour. The chapter concludes with a summary section (*2.4: Summary of Public Space Studies*).

2.1 The Significance of Public Space

This section provides an overall introduction on public space. Its main aim is to establish the significance of public spaces, with a special focus on public spaces in urban settings. This is achieved through an outline of established theorizations of public space, in order to identify important aspects of public spaces as they have been identified in the past half century or so, i.e. during the most recent urbanization process. It identifies the various definitions attributed to public space, which oftentimes prove to be in contrast with one another. A range of dissimilar spaces are identified that fit the different definitions presented, in order to give some concrete, real-world examples of the many manifestations of public space.

Often the most-recognized aspect of public space is its visual aspect, perceived as the image of the city (Lynch, 1960), i.e. public space as the set of physical characteristics that allow us to identify the urban environment around us. However, many other readings of public space exist, and in fact identifying some generally agreed-upon definition of what public space is has proven to be a challenging task. Almost by definition, public space is open to all, and therefore many research fields have approached public space and the activity within it as a topic for research. While

this is very encouraging, and the research and findings that come from it help us understand cities around us and hopefully plan better for the future, it nevertheless highlights the complexity of public space as a topic, and pushes back a definition for it even further. As a starting point on public space from an urban design perspective, Carmona (2010a, 2010b) provides a review of contemporary public space seen through the point of view of multiple urban theories.

Manifestations of public space are identified through some quite diverse and often interconnected aspects: Functional public space (especially in urban environments) is identified as the area allocated to the movement of individuals between private spaces. This type of space includes the streets and sidewalks intended for the movement of people and goods, with a significant part often allocated to motor vehicles, especially in American cities (Loukaitou-Sideris, 1996). This unequal allocation of road traffic at the expense of pedestrian traffic has been a point of criticism against urban planning (Gehl and Gemzøe, 2000), noting the adverse effect road traffic has on human activity (Appleyard and Lintell, 1972).

Economic space is where public space is seen as a driver of economic activity. Such aspects are seen, for example, in revitalization and urban regeneration plans (Roberts et al., 2016). These aim to re-introduce some value in neglected areas, by attracting, for example, private investment (Paddison, 1993) in connection to redesigning and upgrading the quality of urban public space. At a smaller scale, public space is seen as a driver of economic activity not on its own, but rather through its various other properties, which even when applied in absence of public ownership may still drive consumer behaviour, as seen in the global example of the mall (Erkip, 2003) and shopping centre (Lowe, 2005): while such spaces constitute privately owned spaces, they attempt to emulate the experience of public space in a controlled environment (Stillerman and Salcedo, 2012).

Environmental and green space is often seen as public space, as it is often identified in natural reserves, waterways, and wildlife parks, which are often maintained by a regional authority. However, another manifestation of green space that is of interest

to this thesis, is urban open space seen for example in urban parks, which plays a significant role in the sustainability of contemporary cities (Chiesura, 2004, Riddell, 2004, Karlenzig et al., 2007). As one of the few open areas in cities, they are often perceived as being of a communal nature, open to all, even if in terms of ownership that is not, strictly speaking, true.

Social space is identified as the space that facilitates social interaction. This aspect of public space has been theorized as being one of the defining characteristics of urbanity (Larco, 2003), as the space that mediates interactions with the vast majority of people one encounters in everyday life in cities, most of whom are strangers. Additionally, it has been suggested that public space use stems from the social element of public life (Carr, 1992), and furthermore it is the accommodation of such social interaction that constitutes whether a public space is perceived as ”good” or ”successful” (Gehl, 1987, Whyte, 1980, Whyte, 1988). Finally, it is interesting to discuss another point regarding social space that highlights its importance as a defining aspect of public space, that is evident in malls and economic spaces discussed earlier: as Banerjee (2001) notes, a part of social interaction that used to take place in public or ”third” spaces, has, with the perceived decline of public space, moved to spaces that attempt to capture the ambience of one, regardless of whether the space is actually public.

This listing of the different manifestations of public space is meant to provide a glimpse of the multifaceted aspects of public space as an indicator of the immense complexity encountered in the study of public space, and is in no way exhaustive. Within this context then this thesis will approach the study of public space through a subset of its different manifestations, specifically its functional and social aspects. The reasoning behind this selection is as follows: This thesis considers the human as the imperative component of produced space, identifying people as the final consumers of space, and furthermore, this consumption of space is expressed through a person’s physical presence in a space. In other words, this thesis assumes a human-centric approach, as exhibited through a person’s presence in, and interaction with,

a space. Under this approach, functional space is included as the mediator through which people can move through and interact with the space through its physical properties, and social space is included as the feedback parameter which heavily influences how an individual behaves within a space inhabited by *other individuals as well*.

Building on this approach, this thesis considers all social interactions as positive feedback elements in public space use. More specifically, all encounters with others in public spaces will be considered as positive experiences in public life, which add to the experience of being in public. There exist plenty of examples in literature which have demonstrated the opposite effect, and these conflicts of public space are indeed acknowledged by this thesis as well. Examples of exclusionary and repelling[1] social interaction include the avoidance of neglected neighbourhoods, conflicting uses such as skateboarding in public parks and plazas (Woolley and Johns, 2001, Németh, 2006), the presence of ”undesirable” people and activities (Jacobs, 1961, Whyte, 1980; 1988). Furthermore, public space has been the field on which much larger events have taken place, from civil rights movements, to occupations, to demonstrations, which almost by definition introduce an element of conflict. Such events have shaped public spaces to a great degree (Harvey, 2013, p. 73), and continue to affect public space use, through policies and regulations. However, these conflicting and repelling interactions will not be the focus of this thesis, for two main reasons.

First of all, many of the examples outlined above are not inherently related to public space use per se. There are larger issues and conflicts at play in these instances, of a political, social, and/or economic nature, whose resolve materializes in the common spatial environment that is public space. It is not the aim of this thesis to address these issues, and indeed approaching these topics holistically would be a challenging topic even in the complete extent of a work such as this. This work will limit

[1]Repelling interactions are considered those where the existence of one activity drives away another activity altogether

its scope then to such activities and interactions whose realization begins and ends in the public space. As such, it will focus mainly on the physical properties of the activities themselves, i.e. the functional elements of space such as distances, perception and cognition, density and crowding, and not on the subcontexts of these interactions whatever they may be, for example unwelcome/excluded activities, disassociation with certain groups, etc.

The second argument for this approach to social interactions stems from a reading on the elements of public life in urban contexts. Larco (2003), on discussing Jacobs' *The Death and Life of Great American Cities*, notes that *"Great cities are not like towns, only larger. They are not like suburbs, only denser. They differ from towns and suburbs in basic ways, and one of these is that cities are by definition, full of strangers"* (Jacobs, 1961). This role of the stranger is identified as representing two relationships, the stranger as something unknown, and the stranger as something different. It is this element that differentiates dense urban places from other environments, and it is of great importance in this case. Conflicting interactions in public space as defined here are by definition exhibited between groups with different characteristics, i.e. strangers. However, as noted here, it is these interactions that manifest the multifaceted and diverse nature of large cities today. As such, these interactions should and will be considered under the view of a positive encounter, or at the very least a non-repelling activity.[2]

2.2 Studying Human Behaviour in Public Spaces

The previous section established the importance of public space in urban life, as a container and mediator of multiple aspects of cities: Functional space, green space, social space, economic space, among others. Furthermore, it defined urban public

[2]The generalization presented in this assumption has been acknowledged, and was considered at length. The task of including conflicting and exclusionary activities was considered during this work, and would be of great value. However, identifying and categorizing exclusionary and repelling activities in a heterogeneous population of a metropolitan city such as London in a holistic scenario would require a sociological and observational survey far beyond the scope of this work. For this reason, in following sections and chapters, social interaction between individuals in public space will assume an affirmative approach of involved parties.

space as it will be approached in this work, focussing exclusively on its functional and social aspects, and established its overall human-centric approach to the study of public space. Within this context then, it becomes obvious that the examination of public space does not focus on the space itself, but rather on the people that utilize the space, and furthermore on the ways they engage with urban space. The following section will therefore review studies in relevant literature that observed and documented people acting and interacting in space, to better understand how people use public space.

The review of human activity in public space will be divided in three categories, covering movement, stationary activities, and interpersonal distances. Movement and stationary activities are considered here as the two extremes in the full range of human activity in public spaces, i.e. a person presently in a public space will either be traversing *through* the space, or be actively engaged in an activity *in* the space (or anything in between the two), and as such will be approached independently. Movement in urban public space will be considered at multiple scales, both as an activity regarding route planning and wayfinding, as well as in terms of locomotion and physical characteristics of navigating a space. On the other end of the spectrum, observations on stationary activities will be discussed mainly through their spatial footprint and their interaction with the physical properties of the space within which they take place. Regarding the third category, interpersonal distances refer to the observation of specific distances in social interaction. They are highly relevant when considering the interaction *between* people in a space, and will therefore be considered separately.

2.2.1 Movement in Urban Space

This section discusses studies and findings relating to the movement of individuals through space. The aim is to establish an understanding of prevailing theories on how people navigate space. The first part focusses on spatial movement from a neurological approach, discussing navigation and wayfinding through spatial cog-

nition, conceptualization of space, and mental processes involved in path-planning. The second part focusses more on locomotion and the physical act of traversing a space, as observed in urban spaces. In other words, the first section discusses how people navigate *between spaces*, while the second focusses on how people move *within spaces*, identifying this distinction as a matter of comprehension related to scale. This classification in the comprehension of space at different scales has been discussed in literature, with researchers distinguishing between near and far spaces, differentiating such distances as ”perceptual” and ”cognitive”, for example (Canter and Tagg, 1975). In Montello's (1993) scale classification terms, the first section then deals with *environmental* space, a scale big enough that it cannot be comprehended from a single perspective, but rather requires a conceptualization and further mental work in order to navigate, and is the space of buildings, neighbourhoods, and cities. The second part will deal with *vista* space, a space of which the size and majority of characteristics can be apprehended from a single point within the space, given the nature of public spaces often being open spaces as well.

2.2.1.1 Wayfinding and Spatial Cognition

Traditionally, spatial modelling has approached human path-finding through a rational approach, in which it is assumed that path selection is a result of a minimizing process of some defined variable, for example distance (shortest path), time (quickest path), or other cost. It has been argued however that while this approach correlates with observed aggregate behaviour, its applicability to individuals' spatial decisions remains unclear as there might be more factors in effect, particularly psychological and cognitive. It is suggested that people do not read urban networks in absolute metric terms, but rather in geometrical and topological (Hillier and Iida, 2005), which furthermore introduce an element of subjectivity to spatial interpretation, as each individual identifies their environment through their own cognitive functions, creating their own *cognitive map* of the space (Golledge, 1999, Golledge et al., 2000). This in turn introduces distortions on the mental representation of the spatial environment, and therefore it has been shown (Golledge, 1995) that while

individual subjects claim they are using the shortest path in path-finding scenarios, and indeed that path may be the shortest in the individual's mental map of the environment, this does not necessarily correlate to shortest paths in absolute mathematical terms. Overall the field of spatial wayfinding has been increasingly incorporating elements of neuroscience to explore how cognitive functions affect path-planning and navigation.

The continuous study on spatial decision-making and path-planning has identified a number of viable strategies that have been observed to have been used in wayfinding. Spiers & Maguire (2008) list a selection of the prevailing observed wayfinding strategies, including:

1. *Primary Networks* (Pailhous, 1970; 1984), in which subjects rely on a familiar network of main pathways in order to facilitate navigation.

2. *Least-Angle* (Conroy-Dalton, 2003), in which a path is chosen that constantly minimizes deviation from the angle which points directly at the goal.

3. *Fine-to-Coarse* (Wiener and Mallot, 2003), also *Hierarchical Route Planning*, in which it is argued that people plan a route in fine detail that leads out of their current "region", and subsequently plan a route in coarser detail through neighbouring regions, that leads to their destination.

4. *Least-Decision-Load* (Wiener et al., 2004), also known as *Least-Angular-Change* (Turner, 2009), where wayfinding relies on choosing the path that requires the least number of possible decision points, for example following a path until it comes at a right angle to the destination point, then switching to the path that leads directly to the destination.

All of the strategies described above have been observed to be employed during wayfinding (Spiers and Maguire, 2008). It is hypothesized that no single true strategy exists, rather people rely on multiple different strategies, based on knowledge, personal characteristics, etc. What is of further interest here is the fact that cognitive

models of wayfinding proposed based on the above mentioned strategies exhibit a similarity in the overall wayfinding process, observing it as a two-step process: First the planning of the route takes place, followed by the execution of the plan (Spiers and Maguire, 2008). Subprocesses within this overall structure exist in a hierarchy and are executed sequentially and iteratively.

2.2.1.2 Locomotion and Human Pedestrian Movement in Urban Spaces

Regarding group sizes first of all, previous studies (Jazwinski and Walcheski, 2011, Willis et al., 2004, Costa, 2010, Whyte, 1988) suggest that people utilising public/open spaces are most often found to be in small groups rather than on their own, by a noticable amount. Furthermore, it has been observed that group sizes are between two and five people, with two-person groups being by far the most common.

Velocity: Average velocities and movement speeds have been found to be generally consistent across multiple studies, with an agreed average movement speed observed to be approximately 1.5 m/s. Furthermore, movement speed has been observed to be affected by group size, with speed exhibiting an inverse correlation to group size (i.e. larger groups move slower). (For an extensive study on pedestrian speed, see Ishaque and Noland, 2008, also Jazwinski and Walcheski, 2011, Willis et al., 2004, Costa, 2010, Whyte, 1988)

Trajectory: Concerning movement trajectories, literature suggests that the main objective when moving through open/public spaces is distance minimization. Some studies suggest that people will follow the most direct available route to their destination, especially when their goal/final destination is in sight, in which case they will steer directly towards it. Finally, no correlation has been observed between trajectory and group size, potentially suggesting that path planning heuristics (the shortest path approach) in open areas are consistent between people. (Gärling and Gärling, 1988, Jazwinski and Walcheski, 2011, Bitgood and Dukes, 2006, Gehl, 1987, Whyte, 1988)

2.2.1.3 Summary of Movement in Urban Space

A summary of findings so far: Spatial cognition relies on the use of cognitive maps, conceptualizations of space that allow people to plan and perform path-finding tasks. Many strategies to path-finding exist, all equally valid. An agreed-upon characteristic of path-finding is that it constitutes a two-step process, starting with the planning step, and followed by the execution step.

Concerning locomotion and movement at the smaller scale, pedestrian movement speeds in urban environments have been observed by different researchers to be between 1 and 2 m/s, with a mean value of approximately 1.5 m/s (Ishaque and Noland, 2008). People in public spaces are often found to be in groups, rather than solo, with pairs being the most frequently observed group size. Furthermore, and in connection to the observation on speed, people in groups tend to move more slowly on average. Regarding height change, pedestrians tend to avoid sharp changes in level, apparently preferring a longer shallow slope over a staircase. A final point is made on trajectories, which seems to be agreed upon by multiple sources: pedestrians tend to use efficient trajectories and minimize the overall travelled distance. This seems to be particularly true for open spaces, and more specifically in cases where the goal is in sight.

This final point is of some interest here, as it illustrates a difference between movement at different scales: Literature on path-finding and navigation aspects of movement agrees (Golledge, 1995) that minimizing path distance is often reported by subjects, but rarely observed. This can be attributed to limited knowledge and differences between perceived and actual distance. Conversely, on aspects of locomotion and movement in smaller spaces (i.e. Montello's (1993) vista space), the shortest path seems to be the prevailing movement strategy.

2.2.2 Stationary Activities in Urban Space

In this section, the literature review turns its focus on existing research covering human spatial behaviour regarding spending time in public, with a specific focus on stationary activities. The work of two prominent researchers is studied extensively here, that of American urbanist William Hollingsworth Whyte, and Danish architect and urban planner Jan Gehl, both of whom applied a human/user-centric approach in their studies of public spaces, focussing their studies on how spaces are ultimately used by their visitors. Their observations offer an invaluable record of human spatial activity, which can be further expanded upon in order to study how the design of spaces affects their use.

The following sections present some interesting findings on crowd behaviour in public spaces, focusing on stationary activities. Main sources for this type of data are direct observation studies and surveys of such places, carried out by researchers and urban planners in attempts to identify quality indicators for public urban spaces. Such studies offer some interesting insight into peoples behaviour in public, as sometimes records indicate behaviour different from expected. Such surveys had the main research objective of establishing some form of public space quality indicators, and as such were mainly focused on the space, using crowd behaviour as an indicator. However, the codification of crowd behaviour itself, that allowed it to be used as a proxy for quality, can be used in user-centric studies and models as well, and may offer some insight in developing simulations of such spaces.

Jane Jacobs, in *The Death and Life of Great American Cities* (1961), observed a decline of the quality of urban life, and attributed it to orthodox modern city planning and architectural design. Although generally discussing the physical built environment that resulted from the urbanization and rapid expansion in U.S. cities, her work focused equally on non-physical relationships as well, when observing for example that social encounters and relationships at a neighbourhood level have a positive effect on urban life. Along with other observations similar to this, it was identified that social aspects of urban life are at least equally important to physical. Nico Larco

(2003) makes this point clearer, by offering a definition of an urban environment that includes interaction, not form, as a defining characteristic. Furthermore, he identifies urban in sociological terms, as ”the concentration of potential and forced interactions” between individuals, in cases where the density is such that individuals are constantly faced with interactions.

These potential and forced interactions are expressed in the common ground that is the urban public space. Studies have attempted to identify characteristics of space that might affect interaction and social behaviour, and the following sections will discuss observations and findings from such studies. One important note needs to be made here, however, that the studies referenced here were conducted in different cities across the world, but always in countries of the western world. As such, any notes and observations on human behaviour may only hold true in scenarios in western cultures, as different cultures may present different values and perspectives regarding concepts such as personal space and behaviour in public.

2.2.2.1 Different states of moving and standing

Hall (1963, 1966a) identified another interesting characteristic of public behaviour in regard to the perception of personal space in different cultures. In western cultures, when a person is sitting or standing in public they occupy not only the physical volume of their body, but a conceived sphere around them, roughly coinciding with the personal distance zone discussed later. This observation is confirmed by W. H. Whyte’s observations (Whyte, 1980) on people standing in pedestrian flows: Pedestrians would alter their paths to avoid bumping into people standing in their path, or at the very least (surprisingly, to the surveyors) they would apologise, as if they were invading the others personal space. This right to personal space is generally accepted in western culture. Interestingly, however, it is only observed while a person is stationary. If a person is moving, personal distances seem to shrink. Although this remains generally undocumented, it can be observed in contrasting situations, such as crowded sidewalks or train stations, where people on the move form much

denser crowds and brush against one another, while people standing allow for some room between them, keeping densities lower.

2.2.2.2 The 100 percent location

A basic question regarding activities in public space is where people actually situate themselves in space when they engage in a short static activity, such as having a conversation. This was answered by William H. Whyte (1988), when his group were examining standing sidewalk behaviour. The original assumption was that people will move a short distance out of the main pedestrian flow to engage in conversation. Instead, it was observed that people interacting in groups will stay right in the middle of pedestrian flow. A similar observation has been made regarding unplanned interactions between people in other environments such as workspaces, where people tend to interact with one another mainly at areas of high visibility (Sailer et al., 2016). This has been noted by others as well (Gehl, 1987, Ciolek, 1976, in Whyte, 1988, p. 9), and this tendency for people to stay in or very near the main pedestrian flow has labelled such spots as *"the 100 percent location"*. Following this observation, it was hypothesised that some of the most crowded places in stationary activities as well must be street corners, owing to two pedestrian flows meeting perpendicularly. This configuration increases the chances for random encounters, and thus such short interactions seem to cluster around the areas with the most traffic.

Although one might consider that such behaviour would pose a great annoyance to moving pedestrians, it seems not to be the case. When moving pedestrians were observed in the same scenario, Whyte's group observed that people would alter their path to avoid walking into people standing. This observation seems to relate to the notion of personal space, and is presented more thoroughly by E. T. Hall in his work on proxemics (1966a) later discussed in this chapter. Finally, what might be extracted from this observation is that people might perceive others around them as being in a different "state", depending on whether one is walking or standing.

2.2.2.3 On standing in public

This behaviour of standing is observed to change when the survey switches focus, from sidewalks and pedestrian flows to open public spaces. Standing in open spaces is usually associated with a waiting act, for example one might be waiting for an acquaintance, looking up information, or some other activity that is to complete soon. A characteristic edge effect is identified by (Gehl, 1987), when people are observed to stand in open spaces. This is described as the tendency to stand near an edge of the space, such as a wall, facade, entrance, etc. According to Gehl, such spots provide the best conditions for someone to have a good overview of the area, while at the same time minimising exposure. It is further noted that even when such hard edges are not available, people will situate themselves around a feature in the space, such as a column, tree, or lamp, to avoid putting themselves in a situation where they stand out.

2.2.2.4 Seating preferences

The act of sitting is another aspect of social behaviour in public spaces, and is regarded slightly differently than the act of standing, for a few reasons. First of all, the decision to sit somewhere signifies a lengthier duration of the reason for being in that area, for example having lunch, reading, or waiting for an acquaintance that is running quite late. This in turn might enable a person to assess the different options in a space, and since they will be staying in the area for a while, to choose the best option according to their own criteria (such as the least crowded, best view, in the shade/sun, etc). Nevertheless, the number of people sitting in an urban space is generally used as an indicator of the attractiveness of the space, and this process of seating choice might be an explanation of this indicator. [3]

Gehl notes that observed preferences for sitting in urban spaces are quite similar

[3] Further to this, Whyte (1980) has noted and the Project for Public Spaces (2000) has elaborated on additional indicators of successful public spaces: in addition to the number of people sitting in a space, the composition of the group can provide indications of successful spaces. More specifically, successful spaces tend to have a higher proportion of people in groups, a higher than average proportion of women, and also people of different ages.

to the ones observed for standing. He writes: ”places for sitting along facades and spatial boundaries are preferred to sitting areas in the middle of the space, and as in standing, people tend to seek support from the details of the physical environment” (Gehl, 1987, p.157). Furthermore, seats that offer a good uninterrupted view around the area are usually much more preferable as they allow people to see any interesting events that take place in the area. Additionally, such seating arrangements might emerge from people's tendency to maximize access to others while minimizing exposure (Sailer and Psathiti, 2017). This preference might come from the willingness of people to passively participate in social life, and has been observed by Jane Jacobs as well, who writes: ”Large numbers of people entertain themselves, off and on, by watching street activity” (1961, p.45). Codifying this observation, it might be described as such: Given different seating options in an area, with all else being equal, the one offering the best view of the area will be generally more preferable to a person.

Whyte also verifies this observation somewhat, when correlating seating areas and pedestrian flow. He writes (1980, p.33): ”All things being equal, ...where pedestrian flows bisect a sittable space, that is where people will most likely sit.” This generally identifies an entrance (or as close to it) to a space as an attractive sitting space, as it fulfills some of the criteria mentioned earlier as well: It is near an edge or boundary, it offers a good view of the area, and it is near pedestrian flow. It can be safely assumed then that when people decide to occupy a space in public, they will orient themselves towards interesting events in the area, and oftentimes such events can be simply the existence of other people in the area. Or to put it simply, *”people come where people are”* (Gehl, 1987, p. 25).

2.2.2.5 Crowding: emergent order in seating patterns

Whyte further observes a form of collective organisation regarding the seating patterns of people, while studying public spaces in downtown New York. In his work ”The Social Life of Small Urban Spaces” (1980), he provides a detailed descrip-

tion of how people distribute themselves on a single ledge of the Seagram Plaza (Figure 2.1). This ledge had been previously observed as being a favorite spot for people to sit in the area, and a detailed recording of seating patterns was performed, captured through time-lapse photography. The seating patterns were then plotted on a time graph, capturing a typical business day from early morning until late in the afternoon.

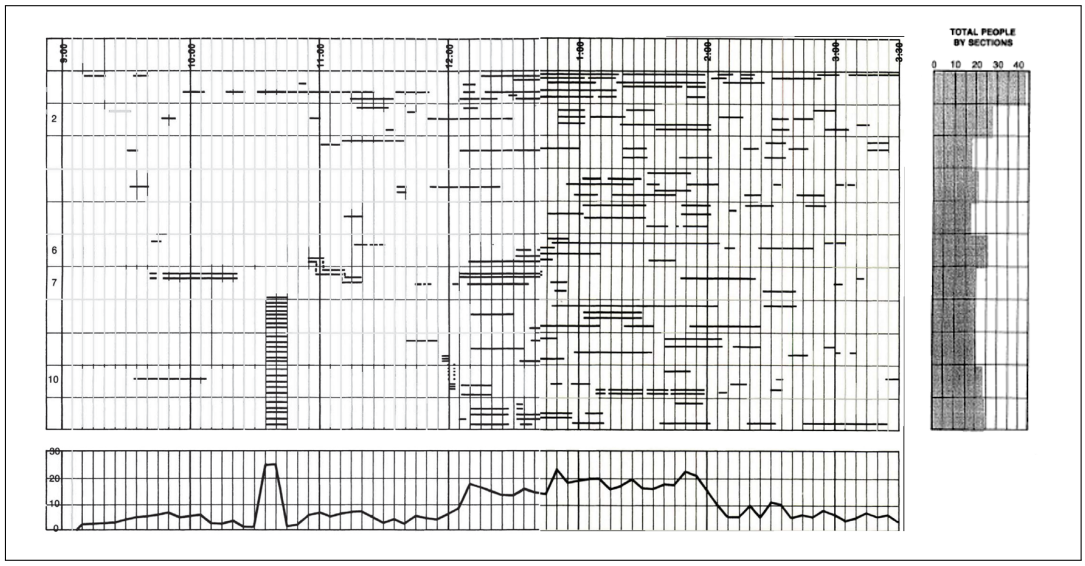


Figure 2.1: Seating patterns over time on a ledge at Seagram's Plaza, New York (Whyte, 1980, p. 70).

In analysing this graph, some interesting observations arise regarding the seating patterns. First of all, as is expected, the number of people sitting is generally low, as is expected in a weekday during business hours. There is a sudden rise in volume around lunchtime, approximately from noon until 2 in the afternoon, as is expected from people having their lunch. During this peak time however, the number of people sitting remains constant, and arguably more interestingly, it is well below full capacity at all times. Furthermore, it is also of note that people were constantly leaving with others taking their place, so this stability in capacity was not due to long-term occupiers. These detailed observations hint at a form of self-organisation, where effective capacity seems to be determined and maintained collectively, through the application of every individual's personal space.

2.2.3 Distances in Social Interaction

Further insight on the way space affects social interaction can be gained from studies on proxemics. The term was coined be E.T. Hall, and was defined as the interrelated observations and theories of peoples' use of space as a specialized elaboration of culture (Hall, 1966a). Its main focus is on the perception of distance between persons engaged in social interaction, and further on the expected behaviours and significance of different distances. Hall studied behaviour and social interaction in mainly western societies, so any interpretation may not apply to different situations, as different cultures may hold different views on personal space and behaviour in public.

Hall identified four distinct distance zones, which are generally obeyed by humans in social interactions. They are labeled intimate, personal, social, and public, and correspond to the level of intimacy between different individuals. These zones coincide with distance zones observed in the animal kingdom as well, with the main differences being observed in reactions to close distances, where for example flight and attack distances seen in animals are largely absent in humans. The four main distance zones are described as follows:

Intimate distance (0-0.45 m): This is the distance between persons where intense feelings are expressed, such as tenderness, love, or anger. Interactions within this distance are not usually observed in public, due to their intimate nature. Also, when this distance is trespassed, individuals feel physically uncomfortable.

Personal distance (0.45-1.2 m): This is the distance for conversations between close friends and family. Friends in small groups in public places will generally situate themselves close to the edge of this distance from each other (approximately 1-1.5 meters). Usually, if a stranger needs to cross this boundary, they will apologise, as a sign of unwittingly invading personal space.

Social distance (1.2-3.6 m): This distance zone holds semi-formal interactions and conversations, usually between acquaintances. Individuals within this distance are

acknowledged as being in a group. The outer zone of this distance is one of the most comfortable distances to keep in public, allowing for interaction, while at the same time not breaching one's personal space.

Public distance (3.6-10 m): This distance zone is used in more formal situations in interactions. In public, this is the distance at which people are acknowledged as being in the same place. Apart from formal situations, people within this distance zone are not acknowledged as forming a group.

The four distance zones presented here constitute observed distances in active social interaction, i.e. between people actively participating in an exchange, roughly corresponding to what Ciolek (1983) classifies as an 'activity'. In addition to this, there exists a form of passive social interaction, relating more to the acknowledgement of other people in space. These spatial aspects of indirect social interaction are recognized to take place within distances of up to 100 meters by some (Gehl, 1987), or 100 yards (roughly 90 meters) by others (Ciolek, 1983), and have been described as 'the social field of view' (Gehl, 1987) and the 'field of co-presence' (Ciolek, 1983). This distance zone is identified as the distance within which people acknowledge others around them as being is the same space, and is generally understood to be limited by the distance over which it becomes impossible to determine personal characteristics of a person (e.g. age, sex, or identity).

More specifically, Gehl describes (1987, p.65) roughly three discrete zones within the social field of vision, labelled here as active interaction distance, spectating distance, and acknowledgement distance.

Active Interaction Distance (0-7 m): This is the distance within which contact and communication between people takes place. Interaction within this distance uses a range of sensory inputs, including aural and olfactory, in addition to the main visual input which at these distances can perceive small nuances and emotional responses. It relates to active communication, for example a conversation, a transaction, and

its limit broadly coincides with the upper limits of Hall's observed public distance, which might stretch up to 10 meters depending on occasion. People being within this distance are generally then understood as being in a group and engaged in an activity.

Spectating Distance (7-70 m): This is broadly the distance range within which people are able to identify other individuals' characteristic, identities, and activities, while not being actively engaged in an activity with them. This zone can be broken down into two sub-categories, near (7-35 m) and far (35-70 m). Near spectating distance is identified as the maximum range within which interaction can take place that includes hearing, although at a limited capacity, for example in a lecture scenario (one-way communication, or possibly a question-and-answer situation). At distances closer to 20 or 25 meters feelings and moods can be perceived. Far spectating distance concerns the distance at which people can be perceived as individuals, and their intentions, actions and activities can be discerned, for example in a sport activity.

Acknowledgement Distance (70-100 m): This is the extent within which figures can be perceived as distinct individuals, and are acknowledged as being broadly in the same space as the observer. No meaningful interaction can take place at this distance. The 100 m mark can be considered as the maximum distance within which social interaction (even passive perception of others) can happen.

This rough classification of spatial extents in social behaviour is identified in works on proxemics as well. Ciolek (1980, 1983) identifies different types of space concerned with the interaction between individuals, primarily at the perceptual level, and secondarily at their spatial manifestation. The first level of interaction is defined as an *f-formation* (Ciolek and Kendon, 1980), and relates to individuals actively engaged in interaction with one another (Figure 2.2). Moving from the interaction outwards, the first order in the typology includes 'r-space', which spatially encompasses 'p-space' and 'o-space'. O- and p-space refers to the area between individuals actively engaged in an activity, along with a buffer zone of personal space

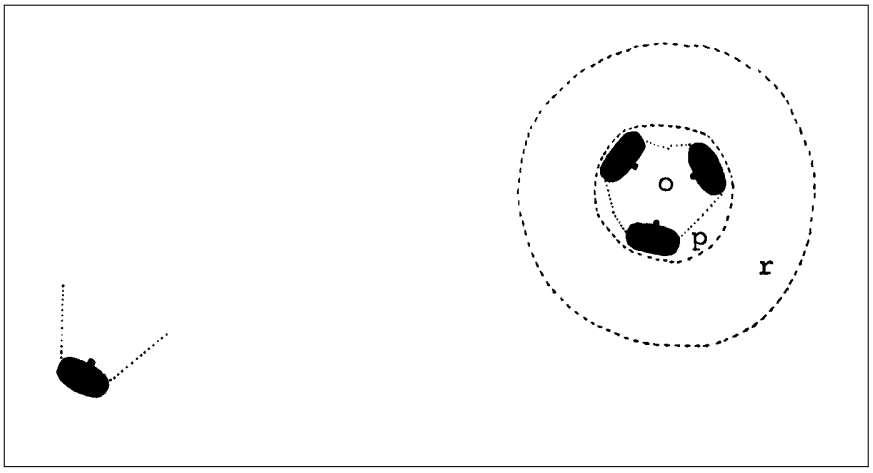


Figure 2.2: An Individual and an F-Formation (Ciolek and Kendon, 1980).

surrounding them. R-space refers to the buffer zone around p-space which defines the ensemble of individuals and activity into a discernible whole.

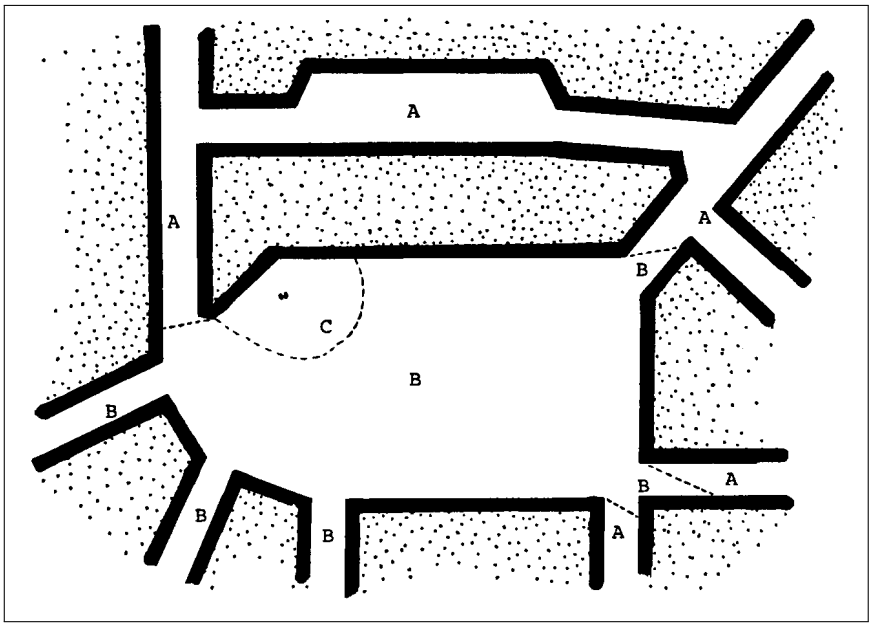


Figure 2.3: Three Zones of Co-Presence (Ciolek and Kendon, 1980).

The second order of typology refers to the space around this group of interacting entities, termed the *spatial zones of co-presence*, and includes a- b- and c-space (Figure 2.3). C-space refers to the immediate space around the group which, although not claimed by the activity, is nonetheless monitored by the individuals, and acts as a transitional space between the group and the rest of the environment. B-space refers to the extent of space around the group that is detectable by the senses,

but has no immediate effect on the group interaction; it is recognized as 'being there'. Finally, the concept of a-space is also postulated to complete the scheme, and refers to the space outside the area detectable by the senses (outside of b-space), it is not detectable, and has no effect on the group interaction.

Although the proxemics lexicon does not generally attach spatial units to space types, the subcontext and perception of the characteristics of each space type allows us to draw similarities with the distance subdivisions in the social field of vision. As such, o-, p-, and r-space can be considered as applicable to Hall's Intimate, Personal, and Social Distance, 0-3.5 m, while c-space corresponds potentially up to the full extent of Gehl's Active Interaction Distance at 7 m, or the full extent of Hall's Public Distance at 10 m. B-space encompasses Gehl's Spectating and Acknowledgement Distances, up to 100m, and by definition anything outside this range is considered a-space.

The classification of distances described above, is by no means exhaustive. However, as the authors note themselves, it can adequately serve as a general reference, when designing with humans in mind.

2.3 Categorization of Human Behavioural Observations

This section revisits the observations discussed in the preceding section through a reformative technical scope. The various notes and observations are restated in a technical definition, in order to begin shaping a formal vocabulary of human social spatial behaviour. These definitions will be implemented in later chapters as heuristics and behavioural rules in the development of computational models of human spatial activity.

Regarding movement: Human wayfinding in spatial environments involves a two step process. During the first step, a formalized conceptual representation of space

is used to plan the overall path as a series of steps between connected locations. This process requires some knowledge of the environment within which the origin and destination lie, and it takes place at a larger scale, thus requiring a mental/conceptual representation of the environment. The second step involves the execution of the plan. It includes the actual movement and locomotion through space, implementing a series of additional subprocesses such as obstacle avoidance, and takes place at a smaller spatial scale, specifically the immediate environment as identified by direct human senses.

Concerning locomotion, the average human walking speed in urban environments has been observed to be approximately 1.5 m/s, and affected by a number of characteristics, such as age, group size, and trip purpose. Sharp changes in level have been observed to be avoided by pedestrians when a better alternative presents itself. Regarding movement within spaces, the majority of pedestrians will move directly towards their goal via the shortest path, if the goal is within vision, and a viable path exists.

Moving and standing are identified as being different states, and are treated differently by others. Specifically, in stationary activities, people claim a larger area around them, which forms part of the activity for the duration. Furthermore, this area is generally avoided by others, and is treated as an obstacle in their course, and unavailable area for any activity.

Regarding the locations of stationary activities including seating and standing, locations seem to differ depending on context. In short activities such as a pause for a conversation, the location is identified to be in the middle of a pedestrian flow, and is more evident in intersection layouts such as street corners. For activities of a longer duration, people seem to gravitate towards edges of a space, such as walls and ledges, or other identifiable features, such as lamp posts, benches, etc., moving away from the pedestrian flow. Regarding seating preferences, people tend to prefer locations close to the edges of a space, if such are available, similar to standing activities. An additional characteristic of seating location preference is observed

in regard to visibility *from* the location, and specifically the number of people and activities visible from a potential seat, as people tend to prefer locations which offer a view of the most activities and other people.

Regarding the observed distances in social interaction in public spaces, the overall functional distance range is identified to be between 0 and 100 meters. Distances of up to 7.5 meters are mainly reserved for acquaintances and people generally involved in a common activity, and potentially smaller (3.5 m) where no additional space is available. Between 7.5 and 35 meters, other people and the activities they are engaged in can be identified and observed, and potential engagements can take place in the form of one-way interaction and communication. Distances between 35 and 70 meters allow for acknowledgement of activities and potentially spectating of any activities that take place, but do not allow for any interaction. Distances between 70 and 100 meters allow for the acknowledgement of other people, and registering them as being in the same space.

2.4 Summary of Public Space Studies

A summary of the contents of this chapter is offered here. Starting with an overview of the many-sided manifestations of public space, the complexity of public space was recognized from the starting point, identifying the fact that no single comprehensive 'correct' definition can be postulated. The human-centric focus of this thesis was established, and by this definition, the functional and social aspects of public space were identified as most relevant to this study.

Following this, a review of existing work on human spatial behaviour in public spaces was presented, discussing important findings of the past few decades. Focus was placed on existing literature that highlights observations on human wayfinding and movement, the use of space and interaction with the built environment within a social context, and the apparent distances observed in human interaction.

The majority of studies discussed in this chapter rely on observation and empirically

gathered data. Finally, the observations were restated in a formal definition, in order to build the technical vocabulary which will inform the rest of this work, and serve as the basis for the rest of this work. As a next step, these observations will be implemented in computational models, which will allow the testing of hypotheses in a virtual environment. Furthermore, the behavioural rules identified here will be embedded in simulations of real-world urban spaces, calibrated to real-time data streams of urban activity, in order to develop real-time spatial simulations of urban activity.

Chapter 3

Computational Models in Urban Studies

This chapter offers a review on models developed for the study of urban systems, with a specific focus on models able to capture human behaviour and interaction in spatial environments. The previous chapter (*Chapter 2: Understanding Public Space Use*) established the importance of public space as the environment mediating a wide range of urban activity, and identified a number of rules observed in human behaviour and interaction in public space. This chapter will discuss the tools and methodologies available in the analytical toolkit that can be of use to the study of Public Space Use (PSU), and will especially focus on methodological approaches aiming at capturing processes as they emerge from the bottom-up.

In order to achieve these aims, this chapter is divided into 4 sections. The first section (*3.1: On Spatial Modelling*) begins with a short review on the evolution of spatial and urban modelling, focussing on a historic overview as well as conceptual and academic spatial modelling aims. In doing so, it will highlight the trend towards capturing systems at ever increasing detail and simulating the dynamic nature of such systems, as exemplified by recent advances in modelling approaches capturing the behaviours of a system's individual constituents, collectively categorised under the term *Individual-Based Models (IBMs)*. A short discussion will follow

on the three main categories of individual-based models, in which the agent-based modelling paradigm is introduced.

The second section (*3.2: On Agent-Based Modelling*) offers a discussion focussed exclusively on Agent-Based Models (ABMs), beginning by discussing definitions, main characteristics, and system applicability of ABMs. Following that, this work presents an overview on the treatment of different aspects in ABMs, more specifically discussing how spatial and temporal aspects are approached, as well as discussing matters of scale. The section closes with a discussion on ABM practices observed in the literature, noting the on-going critique against the lack of proper documentation in published ABM work, and presents different attempts at standardizing or at the very least developing a set of guidelines in the field of ABM.

The third section (*3.3: Applicability of ABM in Public Space Use Studies*) argues how the ABM paradigm can be applied to the study of Public Space Use (PSU). It begins with a literature review of related ABM applications, as identified in the field of pedestrian and crowd modelling, in order to highlight the scope and detail to which spatial behaviour and interaction can be captured at the human scale. Next, it attempts to combine the methodologies presented here, with findings and observations on human behaviour and interaction in public space as outlined in the previous chapter, beginning to form the core for the development of *Agent-Based Models of Public Space Use*, which will be further expanded on in *Chapter 7: Modelling Spatial Behaviour*.

The chapter concludes with a final section (*3.4: Summary of Computational Models in Urban Studies*), which summarizes the content presented in this chapter.

3.1 On Spatial Modelling

3.1.1 Historic Overview of Computational Spatial Models

Although some of the earliest documented geographic models can be identified in von Thunen's agricultural land use model formulated in 1826 (von Thunen, in Hall, 1966b), contemporary approaches in spatial models are generally identified to have their beginnings in the 1950s (Batty, 2008). Models at that time were concerned with predicting future urban growth in terms of transportation, to anticipate and plan for the automobile. Such models were soon coupled with land-use to predict where people would move to based on the distribution of urban functions, giving rise to Land Use Transportation Interaction models (Iacono et al., 2008). Although such approaches were generally found to be too static, with a small set of fixed predictors (land use) used to predict a variable (transportation), and therefore fairly inflexible and coarse to apply to urban planning, they led to the continuing development of more integrated, scaled-down models which could be used in smaller regions (Berling-Wolff and Wu, 2004). Additionally, the development of more efficient computing systems since the 1980s allowed researchers to explore relationships between different actors of urban systems in greater detail: Where previous models worked on the premise of equilibrium, newer approaches allowed modellers to explore feedback between actors and system dynamics by disaggregating entities and exploring change over time. The development of the field of spatial modelling is being actively documented and updated constantly (as offered in the literature referenced in this paragraph, as well as a comparison between different urban modelling approaches by Haase and Schwartz (2009)), and it is not the aim of this work to provide a comprehensive overview of this field. This work is focussed in the most recent advances as seen in disaggregated dynamic models, and the short historic overview presented here helps to provide the context through which these models have emerged. As such, as noted by Batty (2012), the overall trend in the development of computational spatial models throughout the past 50-60 years can be traced:

> *... there has been a sea change from aggregate cross-sectional comparative static models of spatial systems to models that are disaggregate and dynamic.*

From Land Use Transportation Interaction, through spatial econometrics models and systems dynamics models, to IBMs, the above trend can be identified in broad terms. Different modelling approaches often merge and combine for specific applications, and rarely any standards can be identified for any of the approaches discussed here. This modelling timeline presented here loosely follows both the historic as well as the conceptual progress in urban and spatial modelling. As technological advances have allowed for more computing power, and thus for more computationally demanding (i.e. detailed) models, so has a general interest arisen in recent years, regarding the small scale mechanics of systems, which are increasingly being identified as of vital importance to the overall system. This conceptual approach to urban systems analysis is alternatively referred to as a bottom-up approach, where the overall system properties and characteristics are assumed to be largely derived from the actions, reactions, and interactions of its autonomously functioning individual components[1].

The applicability of IBMs to the study of spatial systems and processes has been noted numerous times (Benenson and Torrens, 2004, Batty, 2005), especially when the system of interest involves urban environments, or more specifically human interaction (Heppenstall et al., 2016). The ability of an IBM to capture and simulate emergent behaviour is of notable value for the task of simulating urban/human systems, given the complex nature of such systems. The interest of this work is to study, understand, simulate, and ultimately predict human spatial activity in urban environments. It is evident then that IBMs are a valid tool for this work. As such, in the following section, a review will be offered of the main methodological approaches identified in IBMs.

[1] In contrast to top-down approaches, especially found in planning, where a system is analyzed/designed by focussing proportionately more on the large scale properties

3.1.2 Individual-Based Modelling Approaches

Having traced the spatial modelling timeline and its current branch of disaggregated, dynamic models as identified under the umbrella term of IBM, this section will explore the three prevalent methodologies within this field: Cellular Automata (CA), Microsimulation Models (MSMs), and Agent-Based Models (ABMs). This review will discuss the overall approach of spatial system modelling from the perspective of the individual, and highlight differences between the three methodologies, in order to identify the ones best suited for the purposes of this work, that of simulating users of urban public space.

3.1.2.1 Cellular Automata

The concept of Cellular Automata (CA) was first introduced by von Neumann in the 1940s with further notable work by Wolfram (1984). CA models refer to systems that function based on the discretization of space and time. Spatially, CA consist of a regular grid of cells, which hold discrete values (in their most basic form, CA cells can hold either 0 or 1 values, if functioning in a binary system). The cells all update simultaneously, after calculating their next value according to the values of cells in their neighbourhood, and also the set of rules that describe the model. This discretization of space, time and state makes CA models ideally suited for large-scale computer simulations (Zheng et al., 2009). Furthermore, it has been shown that CA models can be applied to crowd evacuation scenarios, with consistent results between simulations and experiments (Zheng et al., 2009). However, due to their simple approach, CA usually assume homogeneity of the crowd they simulate. Furthermore, they pose a serious limitation, in that they can function mainly in two-dimensional space; otherwise, a three-dimensional space needs to be simplified to a two-dimensional representation in order to provide a grid for a Cellular Automaton. The use of CA in urban and spatial systems has been well documented (Benenson and Torrens, 2004, Iltanen, 2012).

3.1.2.2 Microsimulation Models

The term Microsimulation Model (MSM) is often used as a general term to group disagreggated, dynamic models focussing on individual entities and bottom-up processes. Therefore at times it seems to encompass CA as well as ABMs. When examined through their application to spatial systems (Birkin and Wu, 2012, Wu and Birkin, 2012, Crooks and Heppenstall, 2012), it becomes clear that MSMs are their own category of IBM, distinct from both CA and ABMs. As a rough definition, MSMs simulate individual entities which function in and react to a virtual environment. In contrast to CA, MSMs make a specific distinction between environment and entity, and additionally individual entity definition allows for heterogeneity between entities. Furthermore, MSMs are not restricted to rigid grid space, and can indeed function as a-spatial systems. On the other hand, they have a set of differences to ABM as well. While ABM focus on entity interaction and inter-entity feedback, in MSMs the focus is placed on the effects that environmental changes may have on different types of individuals. In other words, MSMs are focussed on entity-environment interaction, in contrast to ABM where interest is placed in entity-entity interaction as well.

3.1.2.3 Agent-Based models

In the context of this work, Agent-Based Models (ABMs), are autonomous rule-based models which model pedestrians and crowds by simulating individuals as virtual agents. In autonomous models, individual agents are bestowed with rules governing the interaction with other agents in the crowd, as well as with the environment. The rules of behaviour for individual agents are usually implemented in the form of decision trees (Torrens et al., 2012), and pedestrian agents employ a hierarchy of high to low level functions (Pelechano et al., 2007), such as navigation and decision-making (high level), or perception and collision avoidance (low level). ABMs are noted for their capabilities in allowing for a great amount of heterogeneity between individual agents. However, this characteristic of individual simulation

is extremely computation intensive (Bonabeau, 2002, Zheng et al., 2009), and even more so in models that run in real time.

Bonabeau (2002) provides an excellent review of the Agent-Based Modelling framework. First of all, he notes that ABM marks a change of perspective, in that ABMs attempt to describe a system from the perspective of their constituent units. Furthermore, he summarizes the benefits of ABMs as follows: i) ABM captures emergent phenomena, ii) ABM provides a natural description of a system, and iii) ABM is flexible. For the purposes of this work, the first two points provide excellent insight: In i), Bonabeau (2002) argues that ABMs attempt to describe social phenomena, not from a traditional modelling perspective, but with the challenge of reproducing (or growing) them. Furthermore, in ii), it is argued that ABMs are most natural in describing a system composed of behavioural entities, in the sense that agents are the active entities, functioning within the confines of a passive (static) environment, which, at least for the purposes of pedestrian and crowd modelling, provides an accurate description of such a system.

3.2 On Agent-Based Modelling

This section discusses the ABM approach and its applicability in more depth.

3.2.1 Agent-Based Models

An Agent-Based Model (ABM) attempts to model a system as the collection of autonomous decision-making entities called agents (Bonabeau, 2002). The agents function within the confines of their environment, which is the system being modelled, based on their individual assessment of the system and a set of predetermined rules. A loose definition of an ABM may therefore be: *A description of a system, comprising of autonomous entities AND their interactions.*

An ABM consists of two parts, the System and the Component, or otherwise the Environment and the Agent. The relationship between this duality of elements is

the defining characteristic of the ABM approach. On the one hand, the System defines the extents of the environment within which the agents interact, provides input for the agents, and presents aggregate properties for the whole model. On the other hand, individual agents receive input from the environment and other agents and function according to their set of rules, and in doing so provide the overall system its properties.

In this relationship, the different parts (the Environment and the Agents) function on a need-to-know basis. More specifically, the environment does not hold any definitions about the individual agent behaviour (eg. the agent behavioural rules and preferences), and as such cannot calculate the overall outcome of the model. The environment part however can exhibit the overall state of the System, by aggregating the individual Agents properties at any given time step in the model.

In the same manner, it is the agents that contain the definitions for their behaviour, expressed as a decision tree that can be stated in an if-this-then-that form. Individual agents have no knowledge of the system as a whole, but rather rely on their ability to identify their local environment enough to make a decision regarding their next action. This bestows agents with the ability to gather information, and define the ”this” part of the above statement, setting the conditions to which they respond to. Additionally, agents can hold individual preferences which define the ”that” part of the above statement, and allow them to perform different actions in response to their input.

In an ABM, the calculation mechanics are essentially transferred to the smaller entities that are the agents. In doing so, the calculation mechanics (expressed through the agent decision trees) simplify, but at the same time are spread across a large number of actors/agents. This system dynamic can exhibit fairly complex behaviour, even in models with simple agent rules (Reynolds, 1987). It is this characteristic of ABM (simple decisions across multiple actors) that makes them exceptionally good at capturing emergent properties of systems. As such, ABM are becoming the de-facto approach to modelling bottom-up systems, where

> *one models and simulates the behavior of the systems constituent units (the agents) and their interactions, capturing emergence from the bottom up when the simulation is run.* (Bonabeau, 2002).

In the following sections, different elements of ABMs will be discussed in more detail, in order to highlight how concepts about the real world are incorporated in models. More specifically, the next three parts will offer an overview of how concepts of space, time, and scale are implemented. Furthermore, they will highlight the abstractions and definitions that are necessary in order to incorporate such concepts in a manageable and practical manner, and will additionally show how different definitions for these aspects can have a significant effect on model mechanics.

3.2.2 Space in Agent-Based Models

This section will discuss spatial aspects in ABMs, and the different ways physical space is treated in computer simulation. In models that incorporate spatial aspects, inter-agent as well as agent-environment interactions are influenced by the knowledge each agent has about its environment. Such knowledge in ABM is often gathered through an agent's scan of its surroundings, and therefore the methods by which space has been defined (and therefore the way by which an agent gathers information about the world around it) can significantly affect how a model works. The rest of this section will discuss: fragmentation of space, i.e. the existence of a threshold at which space is considered to be indivisible in a model; the number of dimensions used to describe space, i.e. how many dimensions does the model operate in; and finally, ABM approaches that are distinctly a-spatial.

3.2.2.1 Fragmentation of Space

Space in ABMs is often implemented in one of two different ways, depending on the model setup: Discrete Space, and Continuous Space. A comparison through case study between the two approaches is offered by Castle (2011). In the first case, space is regularly fragmented, for example as in a grid, where the division unit is

the default indivisible measuring unit of model space. These approaches borrow heavily from CA, which often function in orthogonal grid space. In discrete space models, distances are measured in multiples of the unit, and any spatial functions are expressed as such, producing fairly rigid spatial relationships between objects. Agent movement and vision is restricted to specific directions; an orthogonal grid for example does not allow movement in 30 degree angles, only in the cardinal directions.

Continuous space models assume space as a continuous property. Distance is expressed using any metric system similar to any used in real-world conditions. This allows for distance unit subdivision and ultimately greater detail in spatial interactions. Any point in model space can be referenced, enabling agents to move in any direction, rotate any amount of degrees, etc. Continuous space models provide more realistic simulations of real-world spatial scenarios, as they allow for much more flexible movement spatial interactions in general.

A third approach to spatial implementations exists, which offers an implicit representation. Such approaches begin from an abstract representation, in which space (often with a specific interest on the movement of individuals in space) is represented as a network of connected locations. In such representations, called *graphs*, important locations are conceived of as points, or nodes, and neighbouring important locations with valid/existing access between them are linked in the representation by a line, or edge. Such implementations of space are notably efficient computationally, and are efficient at representing lattices or similar spatial configurations, i.e. road networks.

3.2.2.2 Dimensionality of Space

Another spatial aspect found in ABMs is the number of dimensions they define their space by. There are examples of ABMs of all three types of space (One, Two, and Three-Dimensional Space), although the vast majority function in Two and Three-Dimensional Space. Examples of One-Dimensional ABM are found in Wolfram's

studies on CA (Wolfram, 1984), where agents exist in linear placement. The spatial interaction in these models is fairly limited, with space being necessarily divided in cells, where each agent has exactly two neighbours to interact with, one on each side. However, when successive states of the simulation are plotted sequentially, the way that order emerges from these simple rules becomes apparent.

Two-Dimensional (2D) models are the most common, as they allow for easy development while still offering a clear description of spatial interactions. Furthermore, in real-world systems, most of the movement and navigation is done on horizontal surfaces, with little to no variation in height, so for the simulation of such systems a 2D representation of space is more than sufficient. In 2D models, system space is a plane on which the agents move and interact, and can be modelled using either discrete or continuous space.

Three-Dimensional (3D) models have only been explored in the past few decades or so (see e.g. Reynolds, 1987), as technological limitations did not allow for the field to expand (as Reynolds mentions (2006), "*The 1987 implementation was an off-line batch process, it took roughly one hour to simulate one second of flocking animation of 80 boids at 30 fps on a then state-of-the-art 1 MHz CPU.*"). Height as the third dimension is addressed in two ways in ABM. One approach incorporates height as a visualisation method, where a model working in 2D space is rendered in 3D (eg. with 2D navigation models, where walls are extruded in the 3rd dimension and agents have height). This allows for much quicker comprehension of model space, as 3D environments are intuitively easier to understand, as seen in (Pelechano et al., 2007). In the other approach, agents interact in 3D space and are able to change height position, for example, as presented in Reynolds' Boids (Reynolds, 1987), simulating the formation of flocks in 3D space. In urban and built environments, 3D models have been developed to incorporate important aspects of the third dimension, such as the effect that falling rubble from earthquakes can have on evacuation scenarios (Torrens, 2014b), and navigation in complex multi-platform (Lin et al., 2013) or uneven varied terrain (Pettre et al., 2005).

A further level of dimensionality, the fourth dimension (4D), is often considered as the dimension representing time. In this work, time in ABMs is considered as a different characteristic and will be discussed separately in a following section (subsection 3.2.4).

3.2.2.3 Aspatial Aspects

In contrast to spatial models, aspatial models are characterized by a complete lack of spatial relationships and the concept of physical space. Aspatial ABM aim to model an abstract relationship between agents, eg. in social interactions, or in simulations of economic activity [citations needed]. In these examples, agents have means other than distance-based to detect and interact with input, eg. in a social simulation, agents might be grouped at random at every iteration, as presented by Wilensky (1997). In aspatial models, basic ABM concepts such as distance and local neighbourhood, which were easily identified and visualised in spatial models, still exist, although applied in different manners.

The concept of local neighbourhood regarding individual agents is altered to fit the model setup as well, as space is not a system attribute. For example, agents might be ranked according to an attribute such as economic output, and be allowed to interact only with other agents within a certain value range. Although in aspatial models, distance as a spatial dimension is not applicable, nevertheless the concept of distance is still useful. This is more evident in models where agent connections are organised in a network graph structure, with agents being the nodes (e.g. in models of social connectivity). In these cases, distance is not space dependent, but rather represents the abstract distance between two nodes, as measured by intermediate nodes in the graph, for example. It is evident then that the concept of distance is still highly applicable, adjusted accordingly to the defining metric of the model.

3.2.3 Scale in Agent-Based Models

Any system of interest can be broken down into the interactions of its individual parts, and therefore examined through the ABM approach, given the loose definition of ABM provided previously. Scale-wise, this means that the agent-based modelling approach provides a scale-free platform for examining systems of interest. Furthermore, it enables researchers to examine scaling properties of the system of interest, or how a system looks and operates at different scales (Batty, 2008). The approach is essentially scaleless; however, specific applications are necessarily defined by their *model scale*.

The Model Scale refers to the system of interests size or extent, and can be defined in two ways: Either by the *environment size*, or by the *agent component size*. By describing a Model by its scale, research focus and design objectives become clearer. For example, by defining the Environment Scale, a Model may focus on a specific area/space and investigate primarily user flows. On the other hand, by defining the agent scale, one is interested mainly in the interaction mechanics between agents, for example one might be looking into developing accurate simulations of cell interaction in biological systems (Holcombe et al., 2012), in which case the Environment extents are irrelevant.

3.2.3.1 Scale Definitions

By defining one of the two main parts' scale of an ABM, its design focus becomes clearer. However, as stated previously, it is the relationship between the two parts (Environment and Agents) that is the core mechanic of an ABM. For this reason, in application, it is necessary to clearly define both size attributes, Environment and Agent Component size. In the following paragraphs, scale and size attributes specific to each Model part (Environment and Agents) will be discussed.

3.2.3.1.1 Environment Scale The definition of Environment Scale (or System Scale) in an ABM helps to establish a scale of reference for the whole model. This

provides the model with maximum and minimum size extents, necessary for deciding elements and phenomena relevant to the system of interest. In models with a clear spatial scope, the Environment Scale often corresponds to the map size as well, and defines the borders of the virtual ”world” within which the simulation runs.

Regarding the necessity for minimum size extents, this is even more evident in large scale models (e.g. regional scale). Without a minimum scale, a large number of elements could be seen as components of the system of interest (essentially, any element observed at a smaller scale, seen as being included ”within” the system's limits). However, most of them are orders of magnitude smaller than the Environment Scale, and their interactions can be better investigated as aggregations into larger components, within the Environment Scale extents.

3.2.3.1.2 Agent Scale The definition of Agent Scale (or Component Scale) in an ABM sets the size of the autonomous components in the model, which in turn establishes the necessary agent attributes relevant to the model focus, and in relation to the model overall scale. As stated previously, agents in their most basic form function via ”if-this-then-that” decisions, responding to changes in their local environment. For this to function, agents need a definition of what their local environment is, and a way of sensing it or gathering information about it in the first place. Both of these requirements are closely related to the agent scale definition. Agent size also determines agent local neighbourhood size, loosely placed somewhere between agent size and environment size. This in turn will determine the agent senses, which is essentially the function to identify and gather information about this local environment. This sensing function is closely related to the model's spatial characteristics. In a-spatial models agents will have local neighbourhoods and senses defined in an a-spatial manner. For example, in a social network model, agents may have a local neighbourhood defined as the acquaintances up to a number of degrees away, and thus their senses function would be in the form of ”averaging opinions on [topic] weighed by node distance”. Accordingly, in spatial models, especially where

agents represent humans, agent sense functions would start to represent some form of vision, and local neighbourhoods would be set to correspond to a person's vision range depending on application (as discussed for example in subsection 2.2.3).

3.2.3.2 Scale Categories

ABMs can be classified into three categories, in relation to their scale. These three categories span the whole range of sizes and dimensions observed in the world around us, and are loosely centered around the human scale as the medium scale. The three categories can be defined as macro-scale (systems too large to be detected by direct human perception), meso-scale (systems identifiable through direct human perception), and micro-scale (systems too small to be detected by direct human perception).

Some examples to make this classification clearer: Macro-scale models may include systems at the regional or international scale, as presented by Parker and Epstein (2011), where the System/Environment Scale can range from national to global, and the Agent Component may represent different cities or nations. Such a model may be investigating transport habits or international trade, for example. Meso-scale refers to systems identifiable at the human scale, and potentially as large as the urban scale. The environment scale can be identified somewhere between the neighbourhood and the city (Malleson, 2012, Malleson et al., 2013) and all examples reviewed in subsection 3.3.1, with the agents representing people or households. Such a model might be investigating crowd flows or land lot uses in a neighbourhood. Finally, micro-scale models focus at systems found at the microscopic scale. Environment scale is around the molecular scale or an organism at the largest, agents represent particles or organism cells, and example models include the investigation of single cell functions or the growth of tumor cells in organisms, for example in the work of Holcombe et al. (2012).

3.2.4 Time in Agent-Based Models

As stated previously, the temporal element is an integral aspect in dynamic models, such as ABMs. Agents have only local knowledge of their environment, and only respond to immediate stimuli, meaning they do not act according to a long term plan. For this reason, simulations advance incrementally, during what is generally referred to as the update function in a simulation. During its update function, each agent collects and processes input received only at the current update, acts according to that input, and then discards all knowledge of the system, until the next update function, where it repeats the same process. In this manner, the simulation advances incrementally, until it reaches a predefined point (usually after a certain time, or when a specific system state has been reached).

There are two ways the passage of time is implemented in ABMs, in much the same way space is implemented, either in discrete time steps, or as a continuous stream of events. The first case is more evident in cases where space is also implemented in a discrete way, for example in CA. In discrete-time models, all agents update simultaneously during a global update function, which counts as one time-step in the simulation. The update function is usually split in two stages, the precalculation stage and the execution stage. In the precalculation stage, each agent gathers all available information about its local environment, calculates its next step, and stores it temporarily. Once all agents have finished the precalculations, the execution phase takes place, where each agent executes the steps calculated previously. This two-step update method is implemented in order to avoid agents updating before others have started their calculations, and thus having agents reacting to the wrong data.

Time as a continuous element in ABM attempts to simulate time in a way similar to that in the real world, where the temporal dimension is continuous. Although time-steps are still used in this approach as well, they take place at much faster rates, usually tenths or hundreds in a second, providing a smoother passage of time to observers of the simulation. This allows agents to update in irregular and/or asynchronous intervals, according to individual conditions, thus allowing for a greater

degree of agent autonomy. As an example, in a pedestrian model where both space and time are implemented continuously, an agent might not need to change/update its bearing until it detects an obstacle in its path, while other agents, navigating through obstacles at that point, might be executing course correcting algorithms continuously.

A final note regarding the temporal aspects in ABMs needs to be made, not in regard to time advancement implementations, but to the way temporal elements might affect the virtual environment. This can be seen in more extensive models that simulate real world systems in great detail, where the passage of time affects environmental parameters in the model. In biological models for example, where inter-species interaction is investigated, the passage of time in the scale of seasons may affect environmental parameters such as available resources, or even agent parameters such as birth/spawn rates or metabolic rates. In urban models, time of day, week, month, and so forth may be a model parameter changing periodically, its value directly affecting agent population numbers or agent preferences.

3.2.5 Agent-Based Modelling Frameworks

During the past years of active development, the Agent-Based Modelling approach has been influenced by many and varied fields, due to its application to the investigation of systems in as many fields. As such, a large part of researchers using the ABM approach are not necessarily familiar or well-versed in software development practices (Angus and Hassani-Mahmooei, 2015). Furthermore, during all this time, no single modelling approach has emerged as the single best approach, and the community is still far from accepting a universal standardized ABM development framework (Heath et al., 2009). In addition to the above, there exists a large number of potential pitfalls one can encounter in the development of an ABM (Wooldridge and Jennings, 1998), made all the more precarious given some recent approaches towards more detailed, descriptive models (Edmonds and Moss, 2005), which makes model verification and validation even harder in a systematic way.

This large disparity between applied approaches in ABM has been documented by some researchers (Heath et al., 2009, Angus and Hassani-Mahmooei, 2015), noting the lack of an agreed-upon protocol. However, some examples of attempts towards a standardization for ABM development, verification, and validation techniques do exist. Manley (2013) provides a conceptual framework for the systematic development of an ABM. Notably, it is platform-independent, and is presented as a hierarchical series of questions which the modeller(s) should ask themselves during the initial development (i.e. at the conceptualization stage). The series of questions begin from a very broad scope, moving gradually into model specifics, and should be approached strictly in the sequence presented (from general to specific).

The design questions in Manley's framework are grouped into four sections, listed here in the order they should be approached. *The Observer* section refers to all aspects that sit outside the model itself, such as mission statement, software, audience, and modeller's bias. By addressing questions in this section, the modeller has a grasp of the context within which their model sill sit. *The World* section refers to the virtual environment of the ABM in question, and includes aspects of time, space, interaction with other systems/models, and the rules that apply in this world. By addressing questions in this section, the modeller establishes a well-defined environment, and sets the physical laws of the model world. *The Interactions* section includes model aspects that define the ways agents interact with one another, i.e. whether agents can exhibit physical interaction, how (or even if) they communicate, and whether there exists some form of resource exchange. This section defines the social rules of the model world. Finally, *the Agent* section includes questions about the actual definition of the agent entity, such as agent characteristics and initial values, their decision making process, and their actions. All of these aspects should naturally conform to the model world rules, as defined in the two previous sections. This hierarchical process employs a deductive approach, by requiring the modeller to answer the broadest questions first, and subsequently flesh out the model based on previously established rules.

Another ABM development protocol, which has recently received recognition in the modelling research community, is the Overview, Design concepts, and Details (ODD) protocol. Originally presented by Grimm et al. (2006) and further revised by Grimm et al. (2010), it provides a set of standards regarding the development of agent-based models and simulations. It requires the modeller to define their model using a platform-independent model schematic, and should therefore allow others to reproduce the proposed model in a different environment. The revised 2010 version will be discussed further here.

The ODD protocol requires the modeller to *'always structure the information about an IBM in the same sequence'*. The proposed sequence consists of 7 elements, grouped into three main blocks: Overview, Design Concepts, and Details. *Overview* includes the 'Purpose', 'Entities, state variables, and scales', and 'Process overview and scheduling' elements. This block provides a general overview of the proposed model, what it aims to achieve, and how. By reading only this information, a reader should be able to implement a version of the proposed model in any Object-Oriented Programming (OOP) language. It is important to note that at this point, Grimm et al. specifically ask modellers to refrain from explicitly describing the model in terms of code, as such aspects should only be considered at the implementation stage. *Design Concepts*, both an element and block in ODD, describes the general concepts that are exhibited in the model, such as whether the model addresses emergence, stochasticity, and how agent interaction occurs. The third block, *Details*, includes the 'Initialization', 'Input data', and 'Submodels' elements. It is at this point that the modeller should discuss model implementation in terms of code used, and provide ample information for readers to reproduce the baseline simulations. The overall logic behind the ODD protocol is that model information is provided hierarchically, from the general to the specific, allowing the reader to constantly build on previous information.

3.3 Applicability of ABM in Public Space Use Studies

This section discusses the potential application of ABMs to the study of Public Space Use (PSU). This potential is identified in state of the art applications of ABMs in closely related fields, and their capacity to be extended in such a way as to capture PSU activity. Such closely related applications are identified in the field of pedestrian modelling, where ABMs are used to simulate the movement of individuals. A review of ABM pedestrian and crowd simulations will be presented in the first part of this section, followed by the second part in which a proposal is presented, outlining how existing pedestrian ABM methodologies can be extended to capture and simulate human activity in urban public spaces.

3.3.1 Agent-Based Models of Pedestrian Movement

This section will review relevant work specifically in the field of pedestrian ABMs to identify recent advances and trends. Although the field is still new, a large body of work already exists, due to its application to and interest from multiple disciplines. For this reason this review will focus on work published since early 2000, looking at how the field has evolved in recent years. For comprehensive reviews of pedestrian ABMs from the perspective of different fields, the reader is directed to existing work: Pelechano et al. (2008) offer a review of crowd simulations as examined from a computer graphics perspective, Papadimitriou et al. (2009) offer a traffic-oriented critical review of the field looking at behavioural modelling assumptions, while a most recent review of approaches from all relevant fields in the development of virtual *Streetscapes* is offered by Torrens (2016).

In recent years, one of the primary aspects of crowd models is the number of dimensions used to describe the virtual environment (most often 2D and 3D), as it may have a direct effect both on implemented methods, as well as model visualisation. The majority of models reviewed here were found to implement space in two dimensions (Penn and Turner, 2001, Turner and Penn, 2002, Batty et al., 2003, Helbing et al., 2005, Helbing and Johansson, 2011, Dai et al., 2013, Dias et al., 2014,

Bandini et al., 2014a, Bandini et al., 2014b, Hartmann and Hasel, 2014, Liu et al., 2014, Crooks et al., 2015, Leng et al., 2015, Narang et al., 2015, Crociani et al., 2016, Fang et al., 2016, Song et al., 2016, Pouke et al., 2016). These approaches offer a top-down plan view of the world, with agents moving on a flat plane. As has been discussed previously, this simplified representation is often found to be adequate in capturing the movement of individuals in space, as most interaction takes place on the ground. Furthermore, the reduction to 2 dimensions increases computational efficiency, and allows for quicker implementation. On the other hand, some models were found to be developed in 3D, making full use of the third dimension (Pettre et al., 2005, Haciomeroglu et al., 2008, Sud et al., 2008, Navarro et al., 2011, Torrens, 2012, Lin et al., 2013, Torrens, 2014a, Torrens, 2014b). These models often implement high-fidelity shapes, animation, and textures, allowing for a more realistic view. Furthermore, some spatial configurations might require a 3D model, as they would be impossible to represent in 2 dimensions (e.g. multi-level buildings such as stadiums). The inclusion of the 3rd dimension often results in more detailed models in all regards at the cost of computational efficiency. A third category is identified as well, placed between the two mentioned here: Some models (especially models developed around the mid-2000) have implemented a 2.5D approach to space, with the model functioning on a 2D plane, but visualized using 3D avatars and walls extruded from the floor, for clearer legibility (Lamarche and Donikian, 2004, Shao and Terzopoulos, 2005, Pelechano and Badler, 2006, Pelechano et al., 2007, Durupinar et al., 2011, Moussad et al., 2011). These models aim to bridge the gap between 2D and 3D, by combining benefits of both approaches: more efficient 2D models, with the visual legibility of 3D avatars.

Having discussed the dimensionality of space in ABM implementations, it is interesting to examine how models fragment and codify space as well in terms of discrete and continuous space, as has been discussed earlier (subsubsection 3.2.2.1). Two main approaches are identified in relevant work: discrete grid space (Penn and Turner, 2001, Turner and Penn, 2002, Batty et al., 2003, Bandini et al., 2014a, Bandini et al., 2014b, Hartmann and Hasel, 2014, Wagner and Agrawal, 2014, Crooks

et al., 2015, Leng et al., 2015, Song et al., 2016), and continuous space (Lamarche and Donikian, 2004, Pettre et al., 2005, Helbing et al., 2005, Pelechano and Badler, 2006, Pelechano et al., 2007, Haciomeroglu et al., 2008, Sud et al., 2008, Moussad et al., 2011, Torrens, 2012, Lin et al., 2013, Dai et al., 2013, Dias et al., 2014, Liu et al., 2014, Torrens, 2014a, Torrens, 2014b, Narang et al., 2015, Crociani et al., 2016, Fang et al., 2016, Pouke et al., 2016). It is interesting to note here that almost all of the 2.5D and 3D models reviewed here implemented continuous space, while grid-based approaches were done exclusively in 2D space.

Pedestrian ABMs have been implemented in the simulation of indoor spaces (Penn and Turner, 2001, Pelechano and Badler, 2006, Castle et al., 2011, Zhou et al., 2012), often investigating issues of evacuation (Zheng et al., 2009, Wagner and Agrawal, 2014) and overall navigation in enclosed spaces (Lin et al., 2013), as well as outdoor/urban spaces (Crooks et al., 2015), investigating safety in large events (Batty et al., 2003) and urban-wide emergency scenarios (e.g. earthquakes, as seen in Torrens, 2015). A third category is also identified, in which synthetic pedestrians move in a continuous featureless plane (Helbing et al., 2005, Torrens, 2012, Helbing and Johansson, 2011, Dai et al., 2013) rather than a dense environment, which aim to capture fundamental aspects of pedestrian and crowd movement (Bandini et al., 2014a, Bandini et al., 2014b, Hartmann and Hasel, 2014).

A variety of pedestrian ABM implementations have been able to reproduce crowd behaviours frequently observed in actual crowds, such as queuing and counter-flows (Helbing et al., 2005, Shao and Terzopoulos, 2005, Helbing and Johansson, 2011, Torrens, 2012, Liu et al., 2014, Leng et al., 2015, Narang et al., 2015, Fang et al., 2016), thus providing a more realistic microscopic representation of crowd dynamics. Such models often implement a continuous space approach along with complex perceptual and steering algorithms, with a specific focus on small-scale interaction between pedestrians.

In simulating individual pedestrian movement, literature suggests that humans implement functions at different levels of cognition, which has been implemented in

multiple pedestrian models (Pelechano and Badler, 2006, Pelechano et al., 2007, Yu and Terzopoulos, 2007, Navarro et al., 2011, Torrens, 2012, Torrens, 2014a, Torrens, 2014b). In these examples, behaviours are differentiated between high and low level functions, with high level functions including behaviours such as path-planning, acquisition of information, and communication, while low level functions include locomotion, obstacle avoidance, and the implementation of vision. High level behaviours are used to define and control the overall purposes of the agent, establishing the agent's strategy, while low level behaviours are used to implement very specific objectives.

Regarding model dynamics, most approaches were found to implement a mostly static environment, with agents reacting to other agents. Indeed this is often the case with models requiring a precalculation of the environment that the agents populate and make use of, and is thus too computationally expensive to re-calculate at each update during the simulation. While such models are good at capturing specific scenarios, their application to other (even highly related) scenarios requires the environment to be set up again, i.e. the agents can not respond to dynamic changes in the environment. Some approaches have aimed at incorporating dynamic changes, through controlled changes in the environment (Pelechano et al., 2007), fully cognitive and reactive agents (Sud et al., 2008, Torrens, 2014b, Crooks et al., 2015) which allow agents to respond to any change in their environment such as dynamic/moving obstacles, or by recording environment states in an efficient manner (e.g. floor fields (Hartmann and Hasel, 2014)). These approaches to dynamic models often employ a more distinct bottom-up approach, in which more cognitive power is given to the agents, along with the behaviours necessary to respond to more varied scenarios. On this topic, it is interesting to note that increasingly models are implementing some form of agent vision (Lamarche and Donikian, 2004, Moussad et al., 2011, Liu et al., 2014, Torrens, 2014a, Torrens, 2014b, Torrens, 2015, Crooks et al., 2015), allowing agents to function with increased autonomy.

Regarding the actual movement of individual agents, as discussed earlier, behaviour

is often categorized into high and low level functions, with high level functions controlling overall path planning, while low level functions control steering and obstacle avoidance. In path planning, the environment itself carries some information on its continuity allowing agents to identify which locations are connected and thus traversable, through the use of Visibility Graphs (Penn and Turner, 2001, Turner and Penn, 2002), navigation meshes (Lamarche and Donikian, 2004), or floor fields (Hartmann and Hasel, 2014). This layer of information is then used by the agents to plan their path using a range of algorithms, including Randow Walk Algorithms (RWs) (Penn and Turner, 2001, Turner and Penn, 2002, Torrens, 2012), Shortest Path Algorithms (SPAs) such as A* and Dijkstra's (Batty et al., 2012, Pettre et al., 2005, Haciomeroglu et al., 2008, Sud et al., 2008, Dai et al., 2013, Crooks et al., 2015), or a form of hierarchical spatial structure (Lamarche and Donikian, 2004, Shao and Terzopoulos, 2005). At the lower level of steering, two main approaches are identified: The Social Forces Model (SF) (Helbing and Molnár, 1995) is found to be the most used approach in steering and obstacle avoidance with moving obstacles such as other agents (Helbing et al., 2005, Pelechano and Badler, 2006, Pelechano et al., 2007, Helbing and Johansson, 2011, Dias et al., 2014), as well as extended versions of it, which include additional elements applying repelling and attracting forces to the agent (Dai et al., 2013, Bandini et al., 2014a, Bandini et al., 2014b). The alternative involves the agent actively seeking the optimal path in front of it taking into account others' trajectories, through a form of trajectory extrapolation of all agents in the local vicinity (Lamarche and Donikian, 2004, Moussad et al., 2011, Liu et al., 2014).

It is of note that recent approaches have started enhancing agent fidelity not only by implementing more efficient routing and steering algorithms, but by also expanding into other fields as well, especially the field of psychology. Some models have implemented greater degrees of heterogeneity in their agents' behavioural trees, by implementing for example leader-follower behaviour (Pelechano and Badler, 2006, Crociani et al., 2016, Fang et al., 2016), in which some agents are more likely to follow other agents' lead rather than rely on their own initiative, which has been shown

to be an effective strategy in evacuation scenarios. In a similar vein, other models have implemented psychological and personality profiles to their agents (Pelechano et al., 2007, Durupinar et al., 2011, Narang et al., 2015, Song et al., 2016), investigating how different personalities might behave and affect crowd behaviour.

3.3.2 Extending Agent-Based Pedestrian Models

This section will illustrate how the ABM paradigm can be employed to provide insights into the study of public space use. The main argument posits that ABM offer a platform for testing phenomena and behaviours in systems in which the existence of complex behaviours and dynamics have been identified. Additionally, ABM work in silico, and as such offer an additional benefit to the study of PSU, in that they enable experiments to be executed in virtual environments, where the development and carrying out of experiments in controlled environments would have been otherwise improbable, or even outright impossible. Activity in Public Spaces exhibits both of these characteristics, namely it is composed of all the human actors *and* their interactions, thus being a system with some degree of complexity, and conducting controlled experiments on such systems is often infeasible, due to the number of parameters that might affect a space at any given moment.

Another way to express and describe this combination of ABM and PSU, is by considering how existing computational approaches of crowd and pedestrian simulations can be enhanced by infusing social interaction rules and stationary activities. From the technical perspective, a large number of models exist which model pedestrian and crowd behaviour in space as the flows of individuals. Furthermore, many approaches aim to produce realistic scenarios of street and urban space activity, often termed *Streetscapes* (Torrens, 2016). This section will discuss how existing approaches in the development of streetscapes can be enhanced to include stationary activities and social interaction rules.

As was expressed earlier, the aim here is to illustrate how a crowd simulation can be enhanced by employing social behaviour and interaction characteristics. By review-

ing relevant work, the Agent-Based Modelling approach was identified as offering a suitable platform for developing such simulations, as it has been successfully applied to a wide range of scenarios highly relevant to the work in this project (as seen in the review of ABMs of pedestrian movement, section 3.3). Scale-wise, ABM has been shown to handle systems such as human interaction in public spaces well. Furthermore, the ABM approach allows for the inclusion of a large number of agent behavioural decision trees. This allows for the development of what Edmonds and Moss (2005) call a descriptive model, by incorporating as many verified parameters as possible in the simulation of a system, in contrast to the reductionist approach generally observed in modelling.

Given all of the above, at this point an outline of a model can be shaped that combines pedestrian modelling with social interaction. At its core, it functions as a spatial interaction - pedestrian model, similar to relevant work, such as the models presented by Torrens (2012), for example. Model Agent Components correspond to synthetic humans, with a comprehensive set of abilities. More specifically, agents are equipped with vision functions, relying on their perception of the environment for input. Also, agents are programmed with motion functionality, allowing them to move in a realistic manner throughout the environment. Additionally, agents have a first set of cognitive abilities, necessary for solving problems of a spatial nature, such as path-finding for calculating a path to their target, and obstacle avoidance functionality, for navigating the scene at finer scales. This set of behaviours is often found in pedestrian simulations, and will allow synthetic humans to navigate a scene with a level of realism beyond other techniques.

In addition to the components mentioned already, agents may hold another set of components, this time relating to identifying and interacting with other agents in a social context. Ideas for such components have already been preliminarily formulated, while discussing surveys on crowd behaviour in public. More specifically, agents may hold different states, for example moving state and standing state. Agents may be perceived differently, according to which state they are in at the mo-

ment. For example, when an agent is calculating how crowded an area is, it will count the number of other agents in that area, excluding all those that are currently moving, as it has been shown that the act of sitting is recognized as having a much stronger presence in public. Furthermore, when two agents meet, they may observe the social distances discussed in Hall's work on proxemics, according to their assumed relationship. Furthermore, the psychological edge effect has been noted, of preferring to stand near an edge of feature in a space. This can be incorporated in an agent behavioural tree, by having agents survey an area, identifying all relevant features, prioritising them, and choosing to interact with the best option. Finally, in addition to all the above, agents might have individual crowding thresholds. It has been shown that people in public spaces will obey and maintain comfortable crowding conditions. This can be incorporated in models by having agents assess their crowding conditions in relation to the situation, and acting accordingly (for example, an agent might have an increased crowding threshold when they are in a popular limited area such as a festival, but lower their threshold once they are in a park). By incorporating such rules and others gathered and verified through observation of public spaces, it seems then possible to enhance pedestrian simulations, and start transforming them into simulations of urban life.

3.4 Summary of Computational Models in Urban Studies

Previous chapters established the need for a tool which would allow the testing of hypotheses and scenarios of human spatial interaction in public spaces, and one such potential tool was identified in the general field of spatial modelling. This chapter's aim was then to explore the relevant literature, in order to first establish whether spatial modelling is indeed a fitting analytical approach, and secondly to identify the branches of spatial modelling most appropriate to the simulation of human spatial activity in urban environments.

In order to achieve this aim, this chapter presented a review of computational modelling approaches relevant to the aim of this work, starting with a short review of the evolution and progress in the field of computational spatial models, as identified within the past 60 years or so. During this time, it was established that researchers have moved from macro static modelling to micro dynamic models, as interest has moved to the small scale dynamics of systems. This disaggregate dynamic approach to modelling was identified as fitting to the aim of this thesis, and as such some of the most prominent disaggregate modelling approaches (CA, MSMs, and ABMs) were further discussed.

Among the three Individual-Based Models (IBMs) that were presented, it was decided that the Agent-Based Modelling approach would be the most fitting. This chapter then provided a review of the ABM approach as discussed in the literature, covering definitions, how the modelling paradigm handles various aspects of the modelled system, as well as a review of development guidelines.

Having discussed the ABM paradigm at length, the next section discussed in particular how ABMs can be implemented to study human spatial interaction in urban public spaces. To do so, the most closely related field of study which has made extensive use of ABM was identified, in the field of pedestrian and crowd modelling. A review of recent advances in ABM pedestrian simulations was first offered, to understand the breadth of scope of such applications and establish the potential. The final point was then made, by demonstrating how this potential could be applied to human social and spatial interaction in public urban spaces, by combining and extending ABM pedestrian models with rules for social behaviour, as identified in the previous chapter.

The following chapter in this thesis will discuss Real-Time Data (RTD) and other opportunities identified within the overall smart cities schemes, and their potential in offering insights as to how urban residents and visitors make use of the public urban environment. That will conclude the first part of this work, consisting of the literature review and the forming of the theoretical groundwork. In *Part II*, the find-

ings of this chapter will be further expanded upon under a more technical approach, discussing how said ABMs can be developed in a programmatic environment.

Chapter 4

On Real-Time Data

This chapter discusses the phenomenon of Real-Time Data (RTD), as it has been identified in the advances of (urban) information and communications technologies in recent years. In order to better understand RTD, in the first section (*4.1: Definitions and Context*), an initial approach discusses the different meanings of the term real-time, as it has been presented in literature, and identifies the definition that most closely matches the system of interest here. Following this, RTD is further discussed within its greater context, which is identified in Smart City schemes evident in various metropolitan cities around the developed world, and is furthermore identified as a subset of a larger phenomenon, that of Big Data (BD).

Due to the interlocking and highly related nature of the two terms (RTD and BD), an attempt is first made to untangle the varied facets of Big Data, and subsequently re-identify RTD within the deconstructed definition of BD. As such, the second section (*4.2: On Urban Big Data*) begins with a discussion on various aspects of big data that are of interest in urban analysis. Aspects discussed include the volume, resolution (both spatial and temporal), as well as matters of data capturing and accessibility. Having untangled and presented BD through its different aspects, the next part discusses some of the critical points towards BD, its ubiquitousness, and its apparent scope of application.

The third section (*4.3: Reframing Real-Time Data within the Context of Urban Big*

Data) presents RTD through the scope of individual aspects of BD. It first identifies the BD elements which constitute a dataset as real-time. Following that, it discusses recent innovations regarding RTD datasets, through applications aiming at making sense of the data, as identified in many City Dashboard applications. Finally, the section concludes by discussing applications of RTD in models of urban systems, thus moving from Real-Time Analytics and Visualisations into Real-Time Models.

Following that, the next and final section (*4.4: Real-Time Data in the Study of Public Space Use*) takes the concept of Real-Time Models a step further in the context of this work, by considering the applications of RTD to the study of Public Space Use (PSU). As a first step, relevant datasets are identified, and their potential application is discussed within the scope of urban human activity. Next, connections are highlighted between relevant RTD, and modelling approaches as have been presented in the previous two chapters. With this, a summary of all findings in these first three chapters is offered, along with their connections, and the tone is set for the next part of the thesis, which will offer a discussion on the technical aspects regarding developing *Real-Time Agent-Based Models of Public Space Activity*.

4.1 Definitions and Context

The discussion on RTD opens with an attempt at a definition, beginning by examining the appearance and usage of the term 'real-time' in literature. Following this, by elaborating on the various aspects of the term, and identifying the relationships between them, a working definition of the term 'real-time' in the scope of this work is offered. Next, RTD is placed within its broader context, identified in relevant literature as the concept of the Smart City. This concept is discussed in more detail, first to establish the context within which RTD comes into focus, and second to identify additional and related approaches to understanding RTD.

4.1.1 References in the Literature

First a discussion and categorization of the different meanings of RTD is offered, followed by the definition of RTD used in this thesis. A reading of apparent relevant literature paints a somewhat blurred picture regarding the definition of the term 'Real-Time', as the term is applied to different (albeit related) concepts and practices. More specifically, three different meanings of the term 'Real-Time' are identified in the literature: Real-Time in terms of temporal fidelity, where data exists at a high temporal resolution, Real-Time in terms of publication time, where a dataset becomes available at (or almost at) the point in time it is captured, and therefore it refers to an ongoing event, and Real-Time in terms of computational fidelity and efficiency, where a computer simulation is able to produce output that is at a high temporal resolution, and/or execute at fairly fast update intervals, i.e. with no detectable delays between updates. These three meanings are discussed in more detail in the following sections.

4.1.1.1 Real-Time Resolution

RTD in terms of temporal resolution (**RT-res**), where data is captured at a very high frequency. In this case, data is being captured and stored at very small intervals. In urban studies, this interval is found to be less than an hour, and often less than that, in 15 of 5 minute intervals, and can range down to capturing the exact second or even millisecond of each individual data point. Phenomena captured by this approach can be re-viewed at a later time on a point-by-point basis, replaying the phenomenon 'as it unfolded', or in other words, *in real time*, rather than offering an aggregated summary of the dataset (e.g. daily summaries). This meaning of Real-Time data can be better understood if considered in contrast to aggregated datasets, for example quarterly reviews of economic activity (not real-time), against daily (or even hourly) records of transactions of a shop/business (real-time in resolution, since they provide a highly detailed record of activity). Such datasets can be used to analyze a phenomenon or system at very high fidelity, and enable researchers to

identify its micro-dynamics, but do not allow the output of the analysis to have an effect on the phenomenon/event itself, since the event took place in the past, and has potentially already ended.

Examples of this type of real-time are mainly found in earlier studies due to technical limitations regarding streaming datasets, since recent advances enable the publication of data as it is being captured, thus turning it into RT-pub (subsubsection 4.1.1.2). Specific examples might include recorded video footage (Bandini et al., 2014b), site surveys (e.g. recording pedestrian flow over time, Gehl Architects, 2004) that are published/analyzed after the survey has ended, archived timestamped datasets, as collected e.g. from social media (Becker et al., 2011), as well as brain studies, analyzing detailed readings of emotions acquired through mobile electroencephalography technology (Aspinall et al., 2013).

4.1.1.2 Real-Time Publication

RTD in terms of publication time (**RT-pub**), where data is published at (or close to) the moment of capture. In this case, data capturing and publishing methods have been streamlined at such a degree that a dataset is being made available in a streaming fashion, offering a view of a phenomenon as it is unfolding in the real-world. This type of RTD is by definition also captured in a high temporal resolution. The analysis of these datasets is of particular interest here, as when combined with RT-comp (next item) systems, allows output to be produced quickly enough to potentially have an effect on the phenomenon of interest.

Examples include monitoring systems for critical infrastructure (e.g. in engineering, SCADA), transportation control systems, communications systems, weather and environmental sensors, as well as social media and Web 2.0 (and later) technologies (Roozemond, 2001, Shamir and Salomons, 2008, Zuccato et al., 2008, Barth, 2009, Calabrese, 2009, Lee et al., 2009, Sevtsuk, 2009, Batty et al., 2010, Sakaki et al., 2010, Shi and Liu, 2010, Becker et al., 2011, Calabrese et al., 2011, Kouvelas et al., 2011, Min and Wynter, 2011, Stefanidis et al., 2011, Tao et al., 2012, Kitchin, 2013,

Tallevi-Diotallevi et al., 2013, Kitchin et al., 2015).

4.1.1.3 Real-Time Computing

RTD in terms of modelling fidelity and computational speed (**RT-comp**). This case does not relate directly to observations of the world and their captured temporal fidelity, but rather to the analysis of high-fidelity datasets. Instances where high temporal resolution data is being analyzed often require the analytical methodologies to maintain a similar temporal fidelity and demonstrate it in their output. In such cases, the analysis can be said to run *in real-time*. Furthermore, depending on analytical approach and processing power, the methodologies implemented might impose a heavy computational load, potentially making the analysis run at speeds slower than real time (i.e. in order to calculate one minute of simulated time, the computation might require more than a real-world minute). However, some cases may require the analysis to not only maintain a similar temporal fidelity, but to also honour the timestep, so that a unit of time in computation corresponds to the same unit of time on observations (e.g. a simulation where one second in the real-world corresponds to one second in the simulated world, or less). This reference to real-time is most often encountered in the field of computer science, where efficiency is a key aspect (Roozemond, 2001, Dia, 2002, Aly, 2008, Pollefeys et al., 2008, Shamir and Salomons, 2008, Barth, 2009, Geiger et al., 2011, Min and Wynter, 2011).

A good example here can be seen in traffic modelling: Consider the problem of calculating all vehicle trips in a given street network, for a given duration (say an hour). A trip distribution algorithm can calculate all trips using a computationally efficient approach, and provide statistics for every trip, as well as for each point in the network. However, this approach does not acknowledge dynamics within the model, for example interactions between vehicles, or the effect of accidents and delays, and therefore the computational simulation can not respond to *real-time* conditions. On the other hand, an Individual-Based Model (IBM) can simulate every individual vehicle in the area of interest, and have them navigate to their

destinations, based on the optimal path *as calculated at each point in the simulation*. This approach can potentially take longer to calculate, but it enables the simulated entities to respond to *real-time*[1] conditions.

4.1.1.4 Working Definition of Real-Time

The three instances discussed above are found in literature and are all described through the term 'Real-Time', although they refer to different (albeit related) concepts. They are not mutually exclusive, and in fact the different facets of RTD are often strongly related (eg. RT-pub are by definition RT-res, while RT-comp often incorporate RT-pub and/or RT-res datasets as input and output).

This thesis focusses on data published at the moment of capture, as it is the interest of this thesis to investigate the possibilities of developing simulations that capture urban activity at present. Under this approach, RT-pub will be considered as the dominant aspect of RTD, with the other two aspects viewed as derivatives. More specifically, RT-res is a direct derivative, since data captured in real-time retains the temporal fidelity any time it is re-used, and RT-comp is considered as the applied part of RT-pub, concerned with solving the technical issues of working with RT-pub. Therefore, when the term *real-time* is used in the rest of this work, it will refer to real-time data in terms of publication.

The term 'Real-Time' has so far been discussed mainly in isolation, as it has appeared in the literature. The next section will define and discuss the broader context within which RTD has been established, as the term has been defined in this work.

4.1.2 The Broader Context: Big Data and Smart Cities

Having defined Real-Time Data (RTD) in terms of meaning and applicability, the next step is to discuss the context within which RTD is most encountered. It is mainly in recent years that the term RTD has gained in popularity, around the turn

[1] In this latter example, real-time refers to each entity's current time in the simulation, rather than the current time in the real-world.

of the 21st century (Graham, 1997, Townsend, 2000). This rise came about due to advances in information technology coupled with networked mobile devices, which in turn allowed people to be constantly connected to each other. This change provided a new alternative, in which people could send and receive information anywhere and anytime, or in other words, *in real-time*. It is interesting to note that initial predictions theorized that it would be mobile phones that would bring about the realization of the Real-Time City (Townsend, 2000). Although this particular technology played a huge part, it was the rise of Web 2.0 technologies, and the subsequent access to such technologies through mobile networked devices (smartphones), that ultimately enabled the Real-Time City.

Furthermore, in addition to making the exchange of information easier, it was the further evolution of these approaches into machine-readable information that gave an even larger rise to the real-time concept of a city. This evolution allowed automated devices to become part of the information exchange, and thus widely broadened the spectrum of potential real-time datasets. This constant exchange of all kinds of information between people, devices, and combinations thereof, and the subsequent archival of these interactions, is what ultimately led to the rise of what is today termed Big Data (BD) (Kitchin, 2014, Townsend, 2013). In recent years, cities have been attempting to harness this non-stop stream of Big Data, in order to improve many of their aspects. These approaches, where urban governance and management relies on the rapid analysis of information, have been referred to using many related terms. As Kitchin (2013) describes:

> *Cities which have embraced information and communication technologies [...] have been variously labelled as* wired cities *(Dutton et al., 1987),* cyber cities *(Graham and Marvin, 1999),* digital cities *(Ishida and Isbister, 2000),* intelligent cities *(Komninos, 2013),* smart cities *(Hollands, 2008) or* sentient cities *(Shepard, 2011). Whilst each of these terms is used in a particular way to conceptualise the relationship between ICT and contemporary urbanism, they share a focus on the ef-*

> *fects of information and communication technologies on urban form, processes and modes of living, and in recent years have been largely subsumed within the label smart cities, a term which has gained traction in business and government, as well as academia.*

According to Ratti and Claudel (2016), the top-down approaches to urban governance of the past decades are deemed insufficient for the development of the cities of tomorrow. These approaches are unable to encompass, accommodate, or even comprehend the diverse needs and wants of the billions of individual citizens of the present and future. What is needed rather is a bottom-up approach, where input and participation from informed citizens is taken into consideration in the planning process, as *"There can be no smart city without smart citizens"* (Ratti and Claudel, 2016, p. 148). Furthermore, citizens can empower themselves through data, both for personal betterment (e.g. for smart homes, cars, etc.) as well as in civic participation (through open data, transparency, etc.). This point is made clearer further by Foth et al. (2016) and Hudson-Smith (2014), who frame the discussion around the triptych of smart cities, citizens, and social capital, with the latter being the driving force behind meaningful change in the smart city. Therefore, it is this bi-directional stream of Big Data usage between city and citizen that will enable the cities of tomorrow: from many small-scale sources large datasets are generated, which allow us collectively to understand how cities work, but also inversely, from the vastness of urban datasets, individuals can make use of specific datasets and information highly relevant to them.

For Townsend (2013), the concept of a 'smart city' has not been properly defined yet, and it is still malleable, since for him the question should not be *"What is a smart city?"* but rather *"what do you want a smart city to be?"* (2013, p. 15). However, at its core, he identifies the interplay between three distinct phenomena: First, rapid urbanization, and the acknowledgement of the fact that in the following decades, the majority of the global population will be living in urban environments, with any problems and opportunities this might present. Second, networked people,

and the capability for the instantaneous exchange of information between individual citizens. And third, networked infrastructure (also termed Internet of Things), referring to the huge number (having far surpassed the number of humans connected to the web) of automated devices connected to the world wide web, and the capacity for unsupervised (by humans) exchange of information in real-time. Townsend identifies the second and third as the tools of this particular era, the proper application of which will potentially help solve the problems of future cities as caused by the first phenomenon, and subsequently play a large part in shaping the future smart cities.

Given the above then, it is evident how RTD is a vital component of the smart city of tomorrow, as identified through the larger application and use of urban Big Data. Therefore, in the following section, RTD will be discussed through the examination of its broader context, identified as that of Big Data (BD).

4.2 On Urban Big Data

This section aims to identify and discuss aspects of RTD that are of importance to this work. However, due to the interrelated nature of RTD, Big Data, and smart cities, as discussed in the previous section, it is often hard to distinguish between the three in relevant literature. Therefore, and since RTD has been identified as a subset of Big Data, in this section Big Data will be discussed holistically, in order to identify aspects and properties that apply to RTD, but are often encountered in the literature as applicable to Big Data in general.

More specifically, an attempt will be made to identify different properties of Big Data, both as have been identified and established in literature, and also through a deconstruction and examination of properties deemed relevant to this work. Furthermore, points of criticism on Big Data and smart cities will also be discussed, to highlight potential problems, and help understand proper applications of RTD.

4.2.1 Properties and Aspects

There have been identified three main characteristics of Big Data, (as described in Kitchin, 2014, Kitchin and McArdle, 2016), also described as the 3Vs of Big Data: volume, velocity and variety.

Volume: On the size of Big Data, both in terms of data size (GB and larger), as well as in terms of coverage, aiming to cover all of the system of interest, rather than sampling (number of data points $n = all$).

Velocity: The real-time nature of Big Data is acknowledged as one of its defining characteristics. However, what is usually mainly established is the stream of information, rather than what or when the information refers to. Although Big Data generally refers to 'now' there are multiple cases where data arriving in a streaming fashion refers to past events. This is discussed in more detail in *Section 4.2.1.1*.

Variety: Relates to the different data types, the structure (or absence of) of datasets, and links between them.

In addition to the 3 Vs, this work will discuss additional aspects of Big Data, in order to better understand its nature and how it relates to the world today. More specifically, it will explore the coverage that BD offers, in terms of spatial as well as temporal coverage (what it covers), its sources and how it is produced (where it comes from), and the different ways BD is offered (how it is accessed). Each of these aspects will be discussed as a spectrum of possible states, as they have been identified through their use.

4.2.1.1 Temporality

Regarding the temporal aspects of BD, two characteristics are considered as the defining ones in this work. First, Big Data is often in high temporal resolution, eg data points represent very fine durations. In urban terms this fine resolution can be in hours, minutes, or less, but greatly depends on application (see following paragraph

on temporal units of BD). The second defining characteristic of BD regarding its temporal aspects is its streaming nature: New data is always being generated and published, and there is no downtime regarding data collection.

However, in addition to the two aforementioned characteristics, another temporal aspect of BD is worth discussing within this context: That of the difference between data publication time relative to time of capture, with the two extreme possibilities termed Real-Time Data, and Historic Data. This dichotomy closely follows from the discussion in the previous section, on RT-pub and RT-res definitions of RTD. In this context, Real-Time Data is published immediately after capturing. Essentially data about an event is made accessible concurrently to the event taking place, or as close to that time as possible (near Real-Time). On the other hand, Historic Data refers to data being made available after the event has taken place and been recorded. In this context, historic can refer to anything that is not ”now”, and is highly dependent on circumstances. As a rough working definition: Historic data is data which highlights an event that has passed, and the data cannot be used by interested parties in order to act and affect the event.

Another way of illustrating this dichotomy between Real-Time and Historic Data would be by discussing the differences in temporal units between capture and publication of the data. As a general rule of thumb, the shorter the duration between capture and publication, the more a dataset can be considered to fall towards the Real-Time definition (and inversely, the longer the duration, the more probable for a dataset to be considered Historic). However, this proves to be an inefficient approach when considering actual units of time, as the differences between Real-Time and Historic data regarding their delay of publication time is entirely based on context and application: For example, in urban traffic monitoring and urban movement in general, hour-old data can be considered as historic, as the urban movement cycles/phases work in much smaller durations, eg. the morning rush hour might last an hour at most, therefore hour-old data reflects an event that has already passed. In a policy context however, day-old or even week-old data can be considered Real-

Time, when compared for example to the time-scale of one of the most reliable datasets, the national census. For example, in (Zuccato et al., 2008), waste-water monitoring can be used to derive indicators of drug abuse, monitored daily, therefore the effectiveness of a new policy on drug use can be studied as it is applied and is adopted by the public.

4.2.1.2 Spatiality

Given the spatial nature of this work, it is of importance to discuss the spatial aspects of Big Datasets. As a starting point, within the wide range of datasets classified as 'Big', it is understandable that a subsection might not include any spatial information. Indeed, specific BD sets are often by their nature a-spatial: For example, genetic information datasets do not relate to a specific place by their very nature. However, when considering Urban Big Data specifically (as is the focus of this work), the existence of spatial properties is of great importance, as it offers a broad spectrum of additional information, and furthermore adds to the relationality of the dataset, through its potential to be considered, intersected, and analyzed along with additional datasets via their spatial attributes. Spatial aspects in BD can be seen to vary a lot, both in terms of resolution, as well as accuracy and extraction of spatial information. A short discussion on applications of spatial Big Data capturing and analysis in multiple instances is presented by Gray et al. (2015).

When focussing specifically on the spatial nature of datasets, a wide range of spatial information is identified. First of all, datasets may differ in their spatial attributes regarding the resolution at which the data is published, from fine to coarse. On the one end of the spectrum, in any spatial dataset a datapoint can exhibit very fine spatial information in the form of geolocated coordinates often captured via the use of a GPS-enabled device, and can therefore pinpoint the exact location of the event that was captured. On the other end, datapoints may be aggregated to a coarser aerial unit, ranging from a small neighbourhood, to coarser units such as a city or country.

In addition to resolution, data can vary depending on the method that is used to extract the spatial information and its subsequent accuracy. More specifically, datasets may often explicitly include the spatial information, in the form of geolocated coordinates, as discussed previously. Furthermore, in instances where the data has originated from an immobile source, such as an installed sensor, the location is also known to be fixed in space, and can be amended if an error is detected, therefore it can be considered to be of very high accuracy. Other often-encountered instances of explicitly geolocated datasets include data originating from GPS-enabled mobile devices, such as navigation devices and smartphones, which allow for capturing the location of an event moving in space. The accuracy depends on the quality of the GPS sensor, with high-quality commercial sensors being able to capture an event with an accuracy of 5-10 meters or less. In addition to explicit spatial information, it is often possible to infer spatial aspects of a dataset even when no geolocation information has been recorded, by parsing the content or metadata of the dataset. This is often especially true when the dataset contains information in the form of written words, which can be analyzed for contextual information, thus extracting spatial information from it, for example by scanning for place names. However, these approaches are very sensitive to noise, and may therefore suffer from high inaccuracy.

4.2.1.3 Sourcing

In addition to examining the spatiotemporal content of big datasets, another valid approach to understanding BD is in the examination of its sources. By identifying where it comes from, and how it is being generated, we can better understand the nature of information that is communicated through BD, and how it may apply to cities and urban systems. Kitchin (2013) identifies 3 main source types of BD: *Directed* capturing, *Automated* generation, and *Volunteered* information, also identified by Ratti and Claudel (2016, p. 54) as *ad hoc sensor deployment*, *opportunistic sensing*, and *crowdsensing*, respectively.

Directed capturing of data refers to data generated through direct observation, meaning that a conscious effort has been made to observe and capture a particular data point. In this approach, an entity of interest (for example a person, a location, or an attribute) is the target of active focus in some manner, via which data is being collected regarding any relevant events. This approach follows from data collection approaches of earlier years in almost all sciences, where the collection of any dataset required some active involvement from the collector, by surveying, running controlled experiments, or focussing a sensor on a particular entity, e.g. a CCTV camera monitoring an entry way. In cases of directed capturing, some (if not all) required aspects are known beforehand, potentially allowing for more fine-tuned data collection.

Automated generation of data has mainly emerged in recent years, as a side characteristic of increased automation in urban systems. In this approach, the generation of data is an inherent and automatic function of a device or system, and can often be the by-product of a different process altogether. More specifically, with electronic devices and systems, it is often the case that a process will keep a record of its function, input, and result, along with other metadata, such as date, time, location, etc., which primarily is of importance to the overseer, to make sure that the automated system is performing as desired. However, with a large enough volume of functions and processes taking place, the volume of side data being generated from such processes potentially becomes such that it can offer valuable information on a significant sample. A prime example of such cases is the automated electronic ticketing system in effect in many large cities (eg. TfL's Oyster Cards in London, UK), which keeps track of individual passenger entries and exits in the transportation network, therefore providing a highly detailed image of the movements of a large portion of a city.

Third, volunteered information as a source of BD refers to information provided by users of a service or general members of the public. In this case, people provide information on a specific topic or service. One subcategory of this source of BD

includes active participation from users, as seen for example in crowdsourced mapping platforms (e.g. openstreetmap.org), where individual users map areas they are familiar with, potentially generating highly detailed maps of the world. This approach has also been termed as Volunteered Geospatial Information (VGI) (Goodchild, 2007), when discussing spatial applications specifically. Another subcategory of volunteered information is seen in the use of the internet as a communication platform, where users publish information through micro-blogging and networking platforms (eg Twitter, Instagram, Facebook), often appending a large amount of metadata, along with the content of the communication itself. This approach has been dubbed Ambient Geospatial Information (AGI) (Stefanidis et al., 2011), in relation to the metadata attached to the messages themselves.

4.2.1.4 Accessibility

In this section, matters relating to data ownership and access to datasets are discussed. The degree of accessibility of various big datasets can be seen to exist in a spectrum, ranging from open and free data which is available to any member of the public, to closed/protected datasets, which restrict access to everyone but a very limited number of specific individuals/organizations. More specifically, four broad categories are identified, presented here in decreasing degree of accessibility: Open Data, Publicly Available Data, Proprietary Data, and Closed Data. A similar classification is presented by the Open Data Institute, in their *data lexicon* (Broad, 2015), noting the existence of 3 types of data: Open Data, Shared Data, and Closed Data.

Open data refer to datasets that are available to any member of the public, free of charge, without restrictions on use. This approach to data ownership and accessibility has started becoming more widespread recently, as seen in local and national open data portals, offering access to a wide range of urban datasets. Some approaches still retain ownership of the dataset, but make it available under an Open Data Licence. For an extended discussion on Open data within the broader BD context, the reader is referred to the work of Kitchin (2014). Publicly available data

refers to datasets that are generally available to members of the public, often free of charge, but are potentially limited to their use due to a restricted licence. The Open Data Institute refers to similar datasets as ”Public Access Shared Data”, describing it as *”data that is available to* anyone *under terms and conditions that are not ’open’”*. Proprietary data refers to datasets that are under a restrictive licence, and access is not allowed by default. Access to such datasets might often require a fee, and/or meeting certain criteria. Finally, closed data refers to datasets that are ”protected” from the public, i.e. data that should not be shared with anyone. Examples of such datasets might include security-sensitive information, business-sensitive, or more importantly, personal information.

4.2.2 Criticism on Urban Big Data and Smart Cities

This section discusses criticisms of Real-Time Data and Smart Cities schemes in general, as identified in recent literature. The main points include the quality of Big Data, ethics on the collection and use of Big Data, issues with derivatives of Big Data, and finally a suggestion on how to approach it.

First of all, regarding the quality of Big Data: It has been suggested that the advent of data-rich sources would result in the end of scientific theory due to its redundancy (Anderson, 2008), as more data would highlight more connections and help optimize on this information alone, therefore making the formulation of models and predictions obsolete. While this may indeed be true sometime in the future (if and when data manages to present us with *all the information*), and while Big Data analytics are indeed supporting a large number of functions nowadays, the underlying issues present in all datasets are not adequately addressed. First of all, Big Data is not necessarily more objective data. Even though stream-lined processes allow us to capture a larger portion of the population, and thus increase the sample size significantly, this merely leads to a proportional increase in bias in data captured and data used, as the number of decisions on discarding, manipulating, and cleaning up data increases as well, especially when considering the number of devices

transmitting and receiving data unsupervised (e.g. automated systems, the Internet of Things). Therefore, it has been noted that Big Data contains a large potential for data bias (boyd and Crawford, 2012, Kitchin, 2014). Furthermore, the mere scaling up of volume of data causes an increase in data error and noise as well (McArdle and Kitchin, 2015, Kitchin, 2014), if no additional actions are taken to ensure data quality. Even in such cases however, where quality standards are ensured, the rapid nature of RTD requires that data points are published as soon as possible. Therefore, this results in data being published which may not have been verified by publishing organisations at first at the time of publication, or in other words, RTD and BD may inherently introduce more error and noise.

Second, Big Data has raised key issues regarding the ethics on its use, as exemplified by recent articles and investigations on whether social media user data, sourced by third parties, was used to analyze, target, and ultimately influence voter behaviour in significant ballots[2]. Starting with the capturing of the data itself: data often comes from user-generated content shared through different platforms and contexts such as social media, i.e. not explicitly research contexts. While the volume and meta-data of such interactions has immense value for scholars when used for research purposes, the user is not necessarily aware of this potential use, and therefore may not have provided explicit consent for their data to be used in such a way. Even in cases where data is public (for example when a user chooses to "publish" their content), the potential levels of public-ness that a user's content is subject to might not be fully clear to them. As boyd & Crawford state (2012), *"Just because [Big Data] is accessible does not make it ethical"*, and therefore researchers working with Big Data, especially user generated data, should be aware of the implications of their work on the data source, and address all issues regarding the ethics of using a particular dataset. Furthermore, it has been theorised that current practices involving Big Data are contributing to widening inequality. While the potential of

[2]Namely the UK EU Referendum, 2016 and the US Presidential Elections, 2016. Sources (Accessed: 2018/04/29): `https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election`, `https://www.nytimes.com/2018/03/17/us/politics/cambridge-analytica-trump-campaign.html`

using Big Data may be of great value to all users and recipients of its output, access to such data and tools is not necessarily open to all, either due to cost or technical knowledge. It is important then to acknowledge that while the pursuit for new data rich sources, tools, and methodologies can in theory be beneficial to society in general, not everyone has the means to extract this value, and in fact such pursuit may be widening existing inequalities, as well as creating new digital divides (boyd and Crawford, 2012, Townsend, 2013, p. 12).

Given the potential issues of Big Data discussed here, it is important to identify derivative issues as well, which may arise through secondary applications of Big Data. Hollands (2008) discusses the validity of the term ”smart city” altogether, when considered as the result of applying Big Data analytic tools to cities. Considering the significant investment and capital that is required to support the change to more efficient and self-reliant city systems, cities may be tempted to adopt the term ”smart city” as a label and marketing tool, rather than an attempt for genuinely progressive applications of Big Data and information and communications technology. Furthermore, as noted by Townsend 2013 such cities may be increasingly reliant on software to perform optimized urban functions, and given the *”buggy, brittle, and hackable”* (Kitchin, 2013) nature of software, this may cause cities to ”malfunction” as well. Therefore, it is vital for researchers to hold a critical stance on cities and regions promoting a smart city agenda at first glance, to ensure that Big Data issues on ethics and quality are indeed addressed in a systematic manner.

Framing the use of BD and RTD in consideration of the points here, it is understood then that BD poses an advantage and a risk: On the one hand, its existence along with the tools to handle it has provided us with a much larger window through which to view the world, and therefore provides an additional proxy for systems and phenomena, unavailable until now. On the other hand however, this window still only accounts for a part of the world, rather than the whole. Given the above, it has been suggested (Mayer-Schönberger and Cukier, 2013) that in order to make use of Big Data, the quality acceptance threshold may need to be set lower, to ”good

enough", as due to its size it inherently presents additional error and noise.

4.3 Reframing Real-Time Data within the Context of Urban Big Data

After having discussed individual aspects of BD, RTD will be identified within this context as a subset of BD, by constraining some aspects.

4.3.1 Relevant Properties: Temporality and Accessibility

The most relevant properties of BD regarding RTD are discussed here in more detail. Mainly, temporality is the main characteristic of RTD, and more specifically the minimal difference between publication time and time of collection, in other words, RTD applies to data which, at the time of first publication, refers to an event that is currently taking place. Additionally, accessibility (and subsequently, reliability) is an important factor, since applications that rely on RTD must have guaranteed access to the dataset as soon as it becomes available.

4.3.2 Real-Time Data Analytics: Urban Dashboards

In order to better illustrate the applications of RTD it would be helpful to briefly discuss some examples where urban RTD has been put to use. One of the most prominent uses of RTD can be seen in visualisation and analytics applications of urban datasets, often described under the term *Urban Dashboards*. City Dashboards aim to bring together many varied datasets regarding urban systems in one view, offering a real-time overview of key urban performance indicators, allowing for quick dissemination and easier consumption by a larger audience. Furthermore, they enable people to acquire an overview of multiple urban aspects as provided through RTD, without requiring the need for specialized knowledge of computer science and data capturing methods on the viewer/user side.

Some of the earliest examples of Urban Dashboards emerged in North America, with one of the initial applications being 'CitiStat', developed for the city of Baltimore in 1999 (Perez and Rushing, 2007), which constituted an attempt at using metrics to identify problematic areas within urban management. It was afterwards opened to the public, by launching a website that provided citizens with city operational statistics. Similar approaches soon emerged in other large metropolitan areas in the U.S.A. (Kitchin, 2013, Mattern, 2015).

One such example is the London CityDashboard[3], developed by the Centre for Advanced Spatial Analysis (CASA) at UCL (Gray et al., 2016), which launched in 2012. It visualizes various real-time city metrics, by collecting data from various open data platforms through the use of their Application Programming Interfaces (APIs). It presents real-time information regarding weather, air pollution, traffic and underground service status, as well as feeds from social media pertaining to London, and the service has been extended to include other large metropolitan areas in the UK. Another example of Urban Dashboards is the Dublin Dashboard[4], an analytical dashboard developed by The Programmable City Project at Maynooth University (Kitchin et al., 2015), launched in 2014. It provides information, including both real-time and time series (historic) data, about multiple aspects of the city of Dublin.

A more integrated approach to Urban Dashboards is that developed for the city of Rio de Janeiro, Brazil, at the Centro de Operacoes Prefeitura Do Rio[5]. It was launched in 2010, bringing together data from multiple city agencies, including traffic, emergency services, utilities, weather, as well as broader public information, into a single urban analytics centre. Its aim was to provide more efficient real-time management of the city, by making it easier to combine and analyze critical urban information (Singer, 2012).

[3] http://citydashboard.org/london/
[4] http://www.dublindashboard.ie/
[5] http://cor.rio/

4.3.3 Beyond Real-Time Analysis: Real-Time Modelling

Having discussed widespread applications of RTD analysis, the possibilities of extending the use of RTD are discussed here, through their incorporation into models. The main reasoning here is as follows: First, the existence of a constant stream of new data covering many varied topics, termed RTD, is considered as a given in the current state of affairs. Second, the capturing, manipulation, analysis, and meaningful dissemination of such real-time datasets has been established, as seen in applications of City Dashboards. Third and final, a next step in this pipeline of urban RTD is their use as input to urban models. This can therefore result in a model of an urban aspect that will be running concurrently to the urban aspect itself, and will be constantly providing some form of output relevant to the urban aspect.

Once workflows have been established regarding acquisition, manipulating, and visualising RTD (which constitutes a big bulk of the technical aspects of working with RTD), the question of whether it is possible then to use these datasets in models becomes viable. More specifically, it has been demonstrated in earlier sections of this chapter that current computing power and methodologies are well equipped to handle models and simulations that execute at a 'real-time' (or even faster) timestep. Furthermore, such models and computational methodologies are capable of handling large volumes of data as input, and of producing similar volumes as output, while at the same time maintaining a high temporal resolution. Therefore, it is hypothesized here that such models and approaches would be able to be receiving as input a stream of RTD, with no significant problem, executing a simulation, and providing an output of some form, in a duration short enough to still be considered as 'Real-Time'. To better illustrate this concept of Real-Time Modelling, the following section will discuss how RTD can be incorporated into models and simulations of activity in public spaces.

4.4 Real-Time Data in the Study of Public Space Use

Having defined RTD and discussed its varying aspects within the context of urbanism, this section will elaborate on the potential that RTD has brought to the field of PSU studies. As was shown previously in this work (chapter 2), the study of activity in public spaces often relies on highly detailed records of user activity with minimal noise, both regarding temporal fidelity (i.e. hourly and minute counts), as well as captured activity (i.e. actions performed by users of space, including movement, avoidance, interaction, etc). Gathering such high quality data therefore incurs a significant cost, as data is either captured through direct observation (Appleyard and Lintell, 1972, Whyte, 1980, Gehl Architects, 2004) and thus requires extensive preparation and long work hours, or is performed through automated systems, such as sensors (e.g. the SmartStreetSensor project[6]) which requires the installation of infrastructure. Furthermore, data often covers a very specific time period, outside of which no data is available. Given all of the above, current practices in the study of public space activity make use of *small data*, as noted by Kitchin (2014, p. 46):

> *[t]heir [small data] production ... allows researchers to effectively mine narrow, tailored seams of high-quality data in order to make sense of the world.*

On the other end of the data size spectrum, RTD addresses the issue of temporal coverage by its very nature, as it is constantly being generated. Additionally, it can somewhat mitigate the high cost of capturing small data, especially when considering ambient and volunteered information, as data is either offered by interested parties, or is included in the meta-data of the message itself. However, as has been discussed, this unsupervised approach to data creation often results in more noise being introduced, and therefore Big Data (and by extension RTD) can be seen as offering a different compromise between data availability and quality (more data at lower quality/more noise), which has been argued (Mayer-Schönberger and Cukier, 2013) may be a valid alternative.

[6] https://www.cdrc.ac.uk/news-archive/18073/

This work aims to examine whether this alternative offered by RTD is indeed valid in the study of public space use. By considering RTD as a proxy for a phenomenon (Mayer-Schönberger and Cukier, 2013), it becomes possible to gather data on public space activity collected passively, and subsequently interrogate the dataset to provide an indicator of activity, even where no ground truth data has been collected. Therefore, by additionally capturing information through ”small data” approaches (i.e. direct observation), this RTD approach to PSU studies is considered as complementary and an extension to traditional approaches.

4.4.1 Relevant Real-Time Datasets

Having discussed how RTD can be used in the study of PSU, this section will focus on potential real-time datasets that can be of particular interest in this study. For the time being, datasets will be identified and discussed only in broad strokes and in general terms, as at this point the main focus is still to build the theoretical framework of this work. Actual datasets that are tested and either used or discarded in this work are presented in later chapters (see Chapters 6, 8, and 9), along with the methodologies used to capture each, and a discussion around the applicability of each. Furthermore, as a clarification, the one common characteristic of all datasets discussed here is their real-time nature, both in terms of being published in a streaming (continuous) fashion, as well as referring to an event that is taking place concurrently to the dataset being published.

The most important area of interest here is human activity in public spaces. Although this parameter widely broadens the spectrum for viable datasets, any datasets that relate to human activity on the ground are potentially of interest here. As such, first and foremost, any datasets containing information on peoples' activity in public space can be considered relevant. This might include volunteered data from users of public space themselves, as seen for example in geolocated social media activity originating from public spaces. Additionally, data originating from directed collection sources can be valid, from monitoring the place itself, such as CCTV footage

and sensors, to automated datasets, such as connectivity records for wireless devices in a space. These datasets, among others, can offer direct information on the number of people currently in a space. In addition to datasets relating to direct human activity in a space, other real-time datasets can be used to infer information on human activity. For example, public transport passenger data can provide information on people arriving at or leaving from an area, which might be of interest in a larger urban context, or in cases where an area is well serviced by public transport, and is fairly isolated (meaning that a large percentage of visitors and users make use of public transport, and therefore some information on activity can be inferred from such datasets). All such examples mentioned here can potentially be used to inform and/or validate a study which focusses on public space activity.

In addition to datasets that directly or indirectly relate to human activity, there are other real-time datasets that might contain information on characteristics and parameters that might affect expected human activity in public spaces, and as such should be considered as well. Given the predominantly outdoor nature of public spaces, weather conditions probably play a large part in current activity in a public space. Therefore, weather and climate information should be considered, when studying public space activity. In addition to weather conditions, a parameter that might affect public space activity is the existence of any cultural events of importance taking place, and so scheduled events might be of interest as well, as can be captured from cultural guides or social media. Such datasets as discussed here can hold information on conditions that affect activity in public spaces, and therefore are of interest to this work.

4.4.2 Implementing Real-Time Models of Public Space Activity

This section brings into focus the arguments from this chapter along with the findings of the previous two chapters (Chapter 2: Understanding Public Space Use, and Chapter 3: Computational Models in Urban Studies). More specifically, the discussion here focusses on the introduction of RTD datasets into Agent-Based Models

(ABMs) of PSU, completing the conceptual framework around the development of *Real-Time Agent-Based Models of Public Space Activity*.

In the previous two chapters, the capabilities of ABMs were presented regarding their application to the development of models and simulations of public space activity. More specifically, it was illustrated how existing ABM approaches and applications for pedestrian modelling can be extended to capture *Streetscape* activity (Torrens, 2016), by incorporating behavioural rules and rules of social interaction, as identified in PSU studies. Therefore, at that point, it was established that such models *can* in principle be developed, and can function at a high fidelity, both temporally and spatially. However, their applicability to real-world cases had not been established yet.

In this chapter, a wide range of datasets was identified, which contain information on public space activity, and are furthermore available in real-time, and therefore at a very high temporal fidelity. Given then the computational capabilities of models as discussed previously, it is posited that such datasets as discussed here *can* be used to inform models of public space use. This combination then allows for the general concept models as discussed previously to be applied to specific areas of the urban realm, and as long as there are datasets pertaining to activity in said areas, for these models to turn into simulations of specific spaces. Furthermore, given the ability of such simulations to run very fast computationally, and of the real-time nature of the datasets discussed here, it is also possible for the said simulations to run concurrently with activities in the space they are simulating, allowing for real-time computational analysis of the space. Finally, given the fact that some datasets discussed previously may also refer to future events and conditions (e.g. weather forecasting, planned events), it is also potentially possible for such simulations to run in a predictive fashion, by continuously simulating a state ahead of current time, and therefore continuously providing predictions of near-future activity.

The findings and arguments presented in the past three chapters form the theoretical framework for this work. They presented and discussed findings in relevant

literature, and illustrated how the three different fields discussed here (Public Space Use (PSU), Agent-Based Models (ABMs), and Real-Time Data (RTD)) should be combined to produce *Real-Time Agent-Based Models of Public Space Activity*. In the next part of this thesis (*Chapters 5, 6, and 7*), the technical aspects of this approach will be discussed in detail, to illustrate how such simulations can begin to take shape and apply to real-world scenarios.

Part II

Methods

Chapter 5

Real-Time Simulation Methodologies

This chapter discusses the overall real-time model of public space activity. It will provide a broad description of the model as a whole, highlighting its different components, their interaction, and interdependence, in order to illustrate overall workflow. In its abstract form, the real-time model presented here is considered as two processes in series: An aggregate activity prediction model, and a spatial disaggregation model, with output from the first process being used as input to the second.

The chapter begins with a conceptual description of the overall model in *Section 5.1*, first by discussing the temporal characteristics of the model. Next the different sub-model parts (aggregate-predictive and disaggregation) are discussed together as components in the overall model, along with a discussion of the way validation is incorporated.

The following section (*Section 5.2*) discusses the process of overall activity estimation in more detail. It presents the two predictive models that were considered in this work. Following this, in *Section 5.3* the spatial disaggregation process is discussed. This takes the form of an Agent-Based Model (ABM), calibrated to capture and simulate public space user activity.

In the second to last section (*Section 5.4*) overall model implementation is discussed, covering the development platform, model visualisation and dissemination,

and output. The chapter concludes with a short summary, connecting findings presented in this chapter with the following chapters, highlighting the areas this work will expand on.

5.1 Model Outline

The overall aim of this work is to examine whether state of the art Real-Time Data Feeds and modelling frameworks can support the development of a *Real-Time Simulation of Public Space Activity*. An underlying model will be developed to support such a simulation, with the goal to continuously predict activities and their locations in a space, disaggregated to the individual level. More specifically, a *Real-Time Simulation of Public Space Activity* in the context of this work is defined as a model that can:

1. Accurately predict the volume of human activity in a public urban space at high temporal fidelity.

2. Accurately predict the types of activities taking place in a public urban space *and* the locations of said activities.

3. Perform the aforementioned predictions of activity concurrently with it happening, i.e. *in Real-Time*.

Spatially, in theory, such a model can be applied to the entire continuous extent of urban public space. However, in the scope of this work, public space will be examined in fragments through case studies, by defining specific public spaces and their borders, and examining them as autonomous entities, cut off from their surrounding areas.

Some initial definitions are required, concerning the overall real-time nature of the model. As discussed in previous chapters, real-time is defined as data published in a streaming fashion, and relating to an event or activity that is currently ongoing. In the context of this work, the temporal unit for this duration (or lag) is assumed to

be adequately measured in a minutes-scale, i.e. a dataset can be considered as real-time (and thus relevant) when it is referring to an event that happened up to several minutes ago[1]. Broadly, within the context of this work, the quarter-hour mark will be used as the maximum threshold, i.e. a data point may be considered as real-time if it was captured up to 15 minutes ago.

There exists an inherent limitation under this consideration: Due to the collection methodologies used, the extent of accessibility to datasets, and the collection methodologies employed by services offering the Real-Time Data (RTD), it is evident that any data collected relates to an event in the near-past. If a simulation model is built around the exclusive use of such datasets, it will always be reflecting events that have already happened, i.e. collecting data for the last half-hour allows us to have a clear picture of events up to the current point in time. Considering the time needed for the simulation itself to run (non-trivial, and depending on computational load, might be measured in minutes) it becomes evident that a simulation that starts 'now' and runs based on datasets that include everything from 'half an hour ago' until 'now' is always representing aspects that fall in the past.

This work takes a different approach, one so that the overall model aims to be simulating public space activity closer to real-time. The overall model is split into different sub-models, each pertaining to a different aspect as will be discussed later in this chapter, with sub-models often working in series, i.e. one sub-model feeds into another. This process itself requires some significant amount of time to be run. Therefore, in order for the overall model to be running in real-time, the overall process is tied to real-world time, with some sub-models relating to the near past, and some to the near future aspects of the space. Parts relating to the near future predict values and parameters relating to the activity of interest, whereas parts relating to the near past collect actual data of the activity that took place (Figure 5.1). This splitting and placing of functions in the future and in the past allows the model (the middle temporal point of the overall model, if you will) to be running in tandem

[1] A measurement in the hours-scale would be too coarse for the needs of this work, while a measurement at the seconds-scale would most often result in not data points per observation

with the real-world.

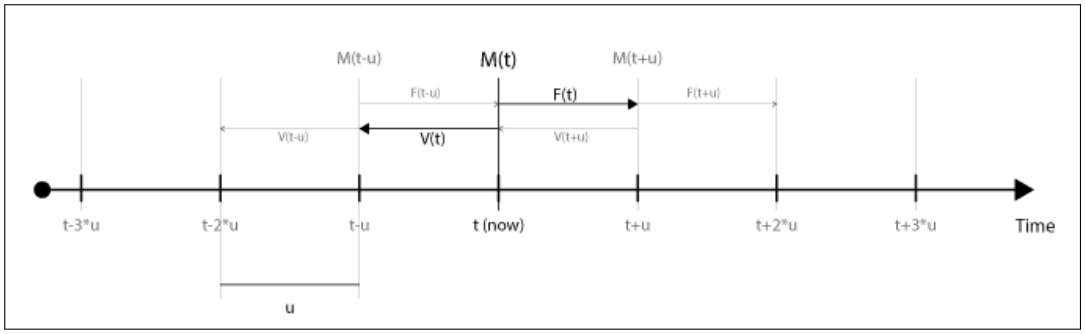


Figure 5.1: Real-Time Model Timeline: Overall model **M** updates at regular intervals **u**. At timestep **t**, the Forecast sub-model **F** predicts the total activity for the following period (from t to $t+u$), while the Validation sub-model **V** collects data on actual activity for previous period (from $t-u$ to t), and compares against the previous forecast of $\mathbf{F_{t-u}}$.

The model of Real-Time Activity in Public Spaces that is presented here consists of two main parts which work in series, and a third auxiliary validation step. For a given point in simulation time, output from the first part feeds into the second part as input. The first part consists of a forecast model for the estimation of overall/aggregate activity in the area of interest, under normal conditions. The second part consists of an agent-based spatial disaggregation model of individual activity.

The first part requires input from multiple real-time sources, all considered as independent variables within the context of this model. It generates output in the form of total activity (a single value for the total current activity in the area of interest). The second part requires input in the form of total current activity as a single value, which it uses to control the agent population size in the simulation, currently and in the near-future. It generates output in the form of individual locations and activities, and density estimations.

The two-step process detailed above provides an estimate of current activity in a space. In addition to these two steps, a third semi-independent step is required for model validation. It acts as a check for both the predictive model, as well as the spatial disaggregation model, providing a feedback loop to recalibrate model parameters. It uses real-time data to check current conditions in the area of interest.

Regarding the predictive model, it reads actual near-real-time data from relevant sources, to validate the output of the predictive model (Figure 5.2). Regarding the spatial disaggregation model, data pertaining to the locations of individuals within the space is required, potentially using sampling methods, and is used to validate the spatial distribution of activities.

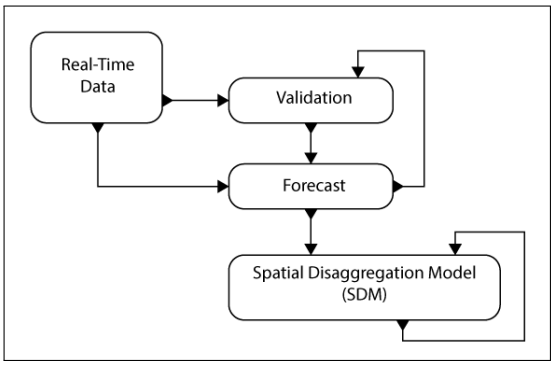


Figure 5.2: Sub-Model Flowchart

The overall model then consists of three distinct parts. It is imperative to discuss the relationships in the model, between the sub-models themselves, but also more importantly between the model and the real-world itself. Two time lines will be considered here: TA, referring to actual real-world time, and TS, referring to simulated time, as used by the model (Figure 5.3).

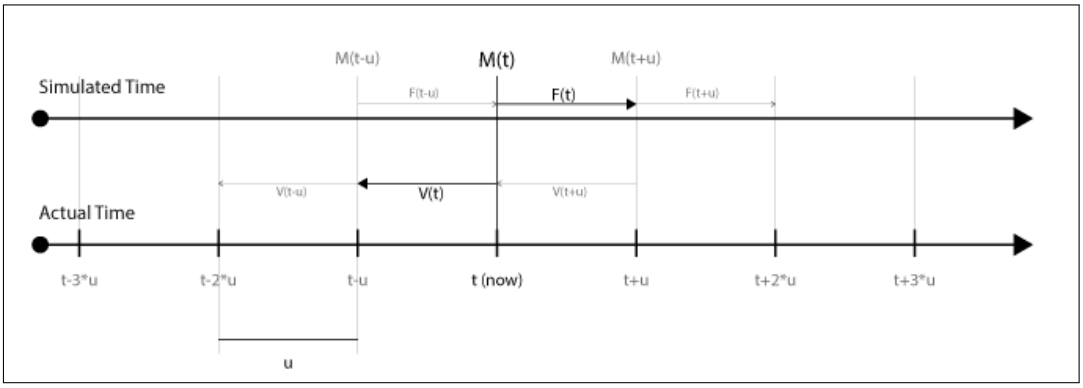


Figure 5.3: Parallel Timelines: Actual and Simulated Time

The first point at which TA and TS coincide is at the start of the predictive model. At time i, the predictive model estimates near-future overall activity for the period between TS_i to TS_{i+1}. This value is then fed into the agent-based spatial disaggregation model, which runs the simulation for the duration TS_i to TS_{i+1}. Note that

no additional real-time data is being collected during the simulation time. Furthermore, the simulation can be run at a faster time that real-time, so that the simulation arrives at TS_{i+1} before TA_{i+1} occurs, it is imperative however that it runs at least in real-time (in computational terms). Also, the data generated during this whole process is all synthetic and predicted data. When actual time TA_{i+1} arrives, the validation sub-model collects all relevant data from RTD sources for the time period TA_i to TA_{i+1}. This data is then compared to the data generated from the model for the corresponding period, TS_i to TS_{i+1}. Any difference between simulated and actual data is recorded. Following this, the model loop starts again, with the predictive model estimating near-future overall activity for the period between TS_{i+1} to TS_{i+2}, taking into account any difference between simulated and actual activity in the previous period, and incorporating that difference as a correction to the new estimation. The new estimate is then fed into the agent-based spatial disaggregation model (Figure 5.4).

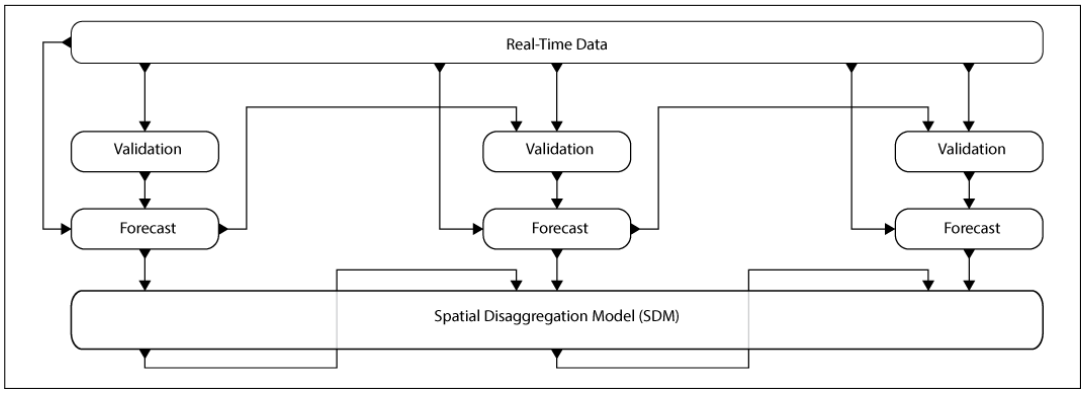


Figure 5.4: Sub-model Flowchart in Continuous Time

It is by using this balance between near-future prediction and near-past collection of RTD, that the model aims to be performing in real-time. It is evident from the description of the model that it is not a 'true' real-time model: it aims to be always predicting the near-future, and once that near-future becomes near-past, to be validating its previous prediction and incorporating it into the next prediction. A 'true' real-time model would require data to be streamed directly into it, from all sources, reliably to the timestep, however if this were achievable for the scope of urban data this model considers, then a real-time model might not be needed at all,

as all information would already be present, and only its analysis would be required.

5.2 Forecast Sub-model

Two different approaches were considered for the predictive model of aggregate activity. In both cases, focus is placed on the activity within the area of interest as the output variable. In both approaches, the aim is to accurately calculate the total aggregated number of people in the space, irrespective of individual visitor characteristics and activities.

5.2.1 Visitor Supply Approach

The first approach considers the area of interest as a receptor of visitor activity. Given that the area of interest is considered as autonomous, this first approach revolves around the idea of capturing the total number of potential visitors arriving in the general area (i.e. just before they potentially engage with the actual space itself). It may draw data from sources such as transport and passenger records, estimating the number of visitors arriving at specific stations, etc. This output is then fed into the spatial disaggregation model (the second part of the real-time public space activity model), where individuals are generated as autonomous agents, and decide on whether they visit the space or diffuse to other local destinations, outside the area of interest, and thus removed from the simulation. This alternative then considers the overall activity model through a supply-oriented approach: It essentially *supplies* the area of interest with potential visitors, who then decide whether to engage in an activity in the public space of interest at a later time. In itself, it does not calculate total simulated activity in the area of interest, but rather it provides a value for *new potential activity*, as it estimates the total number of potential new visitors, with the estimation of activity in the area of interest taking place at a later step (Figure 5.5). In contrast to the next forecasting approach (5.2.2), this model is of a spatial nature, as it takes into account distances of entry points to the area of interest.

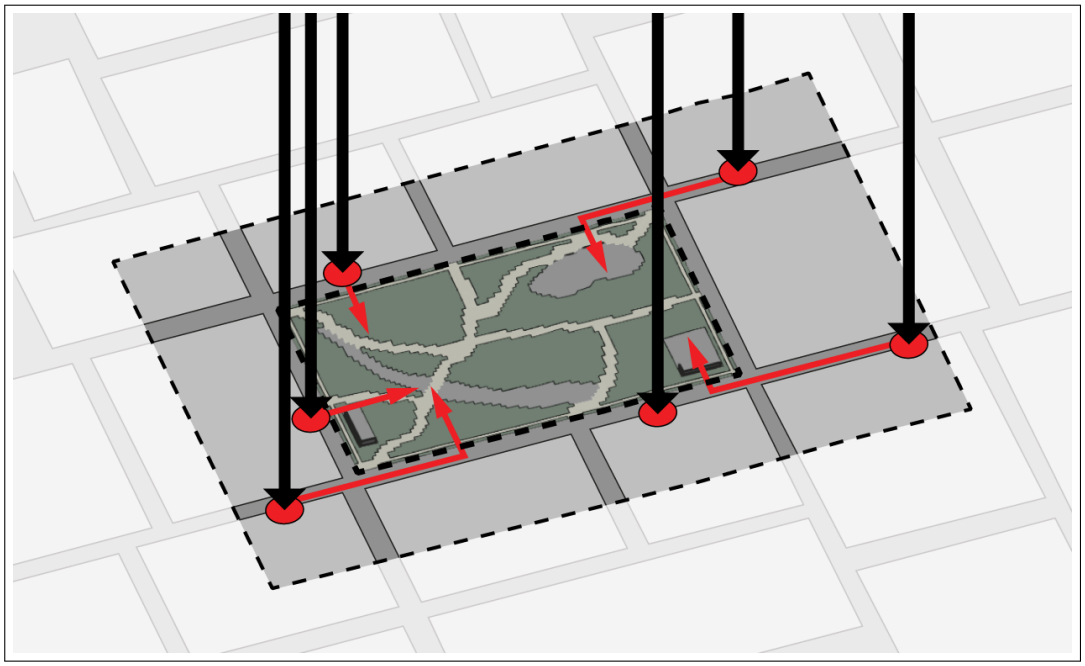


Figure 5.5: Visitor Supply Schematic: For the area of interest (bold dashed line), a buffer zone is created around it (light dashed line), capturing all public transport points in the zone, considered as entry points to the area. A subset of new person arrivals is passed into the area of interest as visitors, and part of the simulation.

The formulation of the model is as follows: The buffer zone around the area of interest captures a set of all potential entry points S, identified as transport network nodes (e.g. bus stops, underground rail stations, etc). For station s in S, at time period t, total passenger exits are E_t^s. Of this total, only a subset is assumed to be directed towards the area of interest, and thus considered visitors. This area visitor volume, denoted V_t^s, is assumed to be affected by distance d_s to the area of interest, so that

$$V_t^s = \frac{E_t^s}{d_s}$$

Therefore, for time t, the total of new visitors N_t from all stations S to the area of interest is defined as

$$N_t = a * \sum_{s \in S} V_t^s$$

with a denoting any additional modifiers. It is important to note that this approach estimates *new* visitors at each update. Therefore, for time period t, total visitor

population P_t in the area of interest is affected by total population at $t-1$:

$$P_t = N_t + b * P_{t-1}$$

where b is a decay factor for the population total at the previous timestep, with a value range[2] $0 < b < 1$. A detailed discussion on the actual estimation is offered at *Section 6.4: Transport Data*, including data sources, methodologies, and results of calculating passenger exits at individual stations at a fine temporal scale.

5.2.2 Total Visitor Volume Approach

The second approach takes the opposite path, by considering overall activity through a demand-oriented concept. In this approach, the predictive model aims to accurately capture the overall number of individuals that are *already engaged in activity within the space*. In this case, the area is treated as an autonomous, cut-off space even more so, as anything that takes place outside the strict borders of the space is completely disregarded. Potential datasets for this approach rely on sources that directly relate to the number of individuals in the area: embedded sensors installed in the area, as well as volunteered visitor data. These datasets inform the model on current activity in the space, and the predictive model aims to accurately predict the total number of visitors that should be in the space at any given point in time. This output value is then fed as input to the second part of the model, the spatial disaggregation model, which converts it into individual autonomous agents, places them in specific locations, and allows them to engage in simulated activities (Figure 5.6). In contrast to the previous Visitor Supply Approach, this model is completely aspatial in nature, as it calculates total aggregate activity based on environmental and temporal parameters.

This model is formulated as follows: Given the nature of the spaces of interest in this work (public, open spaces hosting ephemeral activities), sets of relevant param-

[2]For $b = 0$ the model assumes the population refreshes completely between timesteps; for $b = 1$ the model assumes that visitors never leave the area of interest

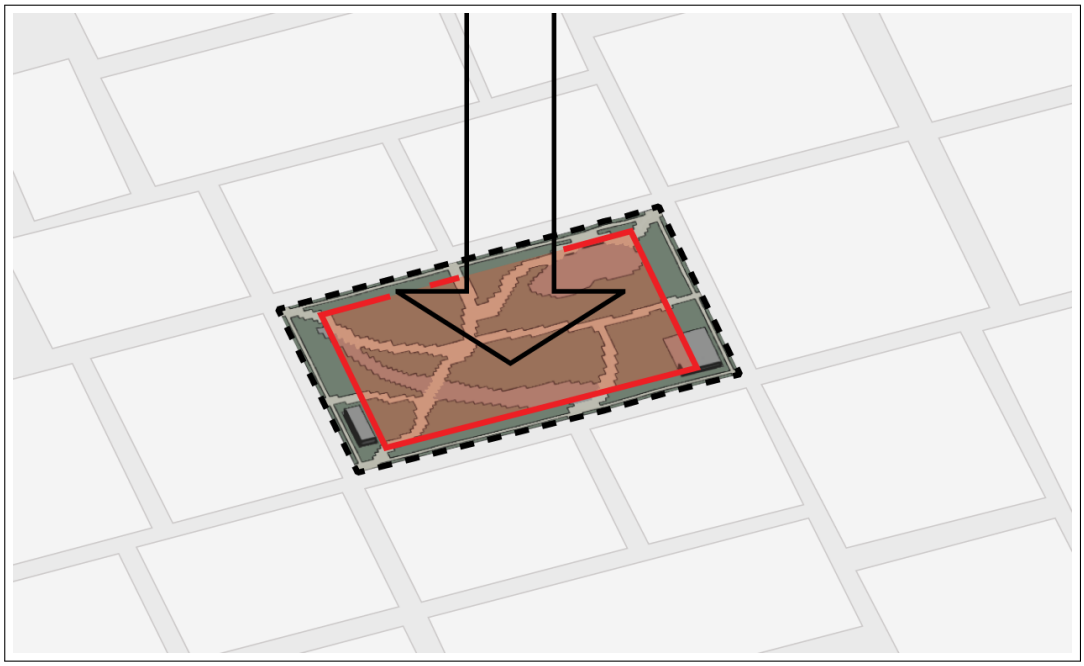


Figure 5.6: Total Visitor Volume: Overall visitor number is estimated as a result of other independent variables, such as time of day, weather, etc. This number is assumed to be the actual visitor volume that will be in the park for the following period.

eters and variables are identified, which are assumed to have an effect on public space activity. These are broadly identified in temporal (e.g. time-of-day, day-of-week) parameters T, weather and climate parameters W, as well as any particular attractions for a given space at a given time, A_t. Therefore, for a given time period t, total visitor population in the area of interest P_t is assumed to be directly affected by the aforementioned parameter sets, so that

$$P_t = T_t * W_t * p + A_t + e$$

where p is a population coefficient, and e is a constant. It is important to note that this approach calculates total visitor volume in the area of interest *right now*, regardless of when visitors arrived at the area. Therefore, in this case, and in direct contrast to the previous approach, it is the estimation of *new visitors at the current time period* that requires the consideration of populations at previous timesteps, so

that

$$N_t = P_t - b * P_{t-1}$$

A thorough discussion on the datasets, methodologies, analysis, and results on this visitor volume estimation approach is presented in *Section 6.1: Online Data - Real-Time Data*.

5.2.3 Estimation Approaches Summary

The first approach (Visitor Supply) was deemed to be too open-ended, requiring potentially multiple independent datasets as input, as it considered total activity as dependent on conditions in the general area. Although this approach is certainly considered as a more accurate/realistic representation given the continuity of urban space, it was found to be inefficient in terms of implementation, calibration, and validation, as is discussed in *Section 6.4.2.3: Disaggregation to minute counts at station*. The second approach (Visitor Volume) is fairly more constrained, as it considers total activity in the area of interest as an independent entity/variable. It assumes a hard-boundaries approach, where the area of interest is cut off from all external influence, and internal activity is considered as an independent, self-reliant element. The Visitor Supply approach was therefore considered as unfitting within the scope of this work. For the remainder of this work, the Total Visitor Volume estimation approach will be used, and all references to Aggregate Estimation Models/Predictions will henceforth refer to the Total Visitor Volume approach.

5.3 Spatial Disaggregation Sub-model

This section discusses the Spatial Disaggregation Model (SDM), which acts as the second step in the overall model. This part of the model is in principle unrelated to any notion of real-time. Its main function is to receive an independent number variable as input, and convert it to spatial activity in the area.

5.3.1 Basic Principles

Three main objectives are identified as vital regarding the real-time simulation of user activity in public spaces, and will be discussed in this section. Two are concerned with the activities themselves, while the third relates to the temporal continuity of the SDM, within the larger context of real-time modelling. Specifically, the activity-related objectives focus on the types of activities taking place in public spaces, and the interactive nature of the activities. The temporal continuity aspect is concerned with the temporal nature of the SDM, when coupled with a real-time forecasting model of public space use. This section will first discuss each of these three model requirements in more detail. Following that, it will identify the Agent-Based Model (ABM) paradigm as a suitable platform for the implementation of the SDM.

5.3.1.1 Relevant Spatial Activity

As a primary requirement, the Spatial Disaggregation Model (SDM) should accurately capture and reproduce individual human activity as identified through its spatial footprint. In other words, the principal objective of this sub-model is to place virtual individuals in the area of interest, with a high degree of accuracy in terms of location. In order to achieve this, the different types of activities that take place in public spaces need to be identified, along with the spatial footprints and characteristics different activities might exhibit, and altogether be incorporated in the SDM.

As discussed in earlier chapters (*Section 2.2: Studying Human Behaviour in Public Spaces*), there is a wide range of human activities observed to take place in urban public spaces. These individual activities have been broadly classified into two categories, moving activities and stationary activities. Therefore, as a broad principal objective, the SDM will aim to capture the movement of individuals through public space, as well as any stationary activities public space users engage in. Such aspects of human spatial activity are expected to be affected by spatial configuration, and

therefore it is expected that the physical environment itself will play a large part in capturing and driving simulated user activity. As such, as a secondary aspect, an accurate representation of the area(s) of interest is needed as well.

5.3.1.2 Visitor Interaction

In addition to the effect the physical environment can have on human spatial activity, there is another aspect that might have a similar (if not greater) effect on individual activity, which is the influence other public space users might have on an individual. Although it has been implicitly discussed, a clearer picture needs to be presented here, regarding human interaction in public spaces. It is generally accepted that humans in social situations do not function completely independently, but rather acknowledge one another, and it has been further suggested that this interaction (even at the passive level of acknowledgement) is one of the defining aspects of urban life (Jacobs, 1961, Larco, 2003).

There are numerous instances in relevant literature where (social) interaction is seen as an important factor in human behaviour, both in empirical/observational studies, as well as in computational/theoretical studies of human behaviour. Furthermore, this effect of interaction seems to be applicable to both moving and stationary activities. More specifically, observational studies on pedestrian movement seem to indicate a correlation between group size and movement speed (Gärling and Gärling, 1988, Costa, 2010), while from a theoretical point of view, a prominent pedestrian dynamics model incorporates interaction as a core element (*Social Forces Model (SF)*, in Helbing and Molnár, 1995). On the other hand, stationary and seating activities in public spaces have been observed to correlate with the existence of other users in the space (Whyte, 1980; 1988, Gehl, 1987). It is therefore important for the SDM to include public space visitor interaction as a core element, rather than assume isolated behavioural heuristics, as it is expected to have a great effect on overall activity.

5.3.1.3 Model Persistence

As stated previously, the disaggregation model in itself is disconnected from any real-world temporal parameters, i.e. it does not relate to real-time conditions. However, its integration with a real-time forecast model raises some questions regarding the potential effect of continuous time on the disaggregation model itself. More specifically, the issue of disaggregation model continuity arises, due to the combination of multiple elements in the overall real-time public space activity model working at different temporal scales/updates. These elements are the update frequency of the forecast model (defined here as approximately 10-15 minutes), public space user activity duration (majority is significantly more than 10-15 min), and the potential interaction between different users of public space.

What is proposed here is a requirement for the disaggregation model to exhibit *persistence*. Persistence is broadly defined as the continued existence of the disaggregation model and its parts/components, for a significant duration of time. This requirement will be demonstrated by considering a discontinuous implementation of a disaggregation model, and illustrating the shortcomings of this approach when applied to simulations of real-time public space activity.

The main function of a basic spatial disaggregation model is to convert an aggregated (single) value into multiple elements/entities, dispersed in space, exhibiting some degree of spatial autocorrelation. Such a model may be coupled with an aggregate forecast model (as discussed e.g. in 5.2.2), executing/calculating a new spatial distribution every time a new scheduled prediction is provided, i.e. is discontinuous. Such a model may indeed be valid for the purposes of this work, if certain conditions hold true: First, the disaggregated entities' temporal relevance must be smaller than the forecast model update, so that spatial distribution of activity has completely refreshed between predictions. This would allow the disaggregation model to calculate a new spatial distribution at every update, given that all entities would be considered as 'new' in the space. Second, entities must have no effect on each other, or in other words, entities must operate under 'blind' autonomous rules,

disregarding any other entities in the space.

On the second condition (entity interaction): when considering public spaces and user activity, it has been observed that human decision making is affected by others' actions, especially when considering activities in public space (Jacobs, 1961, Whyte, 1980; 1988, Gehl, 1987). Therefore, the second condition cannot be considered to hold true regarding human activity in public spaces. The first condition (entities' temporal relevance) partly depends on model parameters: If the predictive model is scheduled to run at a large enough timestep, it can be assumed that between two timesteps, all entities will be different. Regarding activity in public spaces, some minimum values can be considered (Ipsos Mori, 2015a): In parks, typical visit duration is between 30 minutes and 2-3 hours, with average visit durations approximately 80-90 minutes. As has been discussed previously (*Section 5.1, Chapter 4*), within this work, the threshold for RTD is placed at the 15 minute mark and sooner. Therefore, the forecast model can be considered to update at least every 30 minutes, and therefore disaggregated entities certainly persist over multiple forecast sub-model updates (Figure 5.7). Under these considerations, the spatial disaggregation model requires a continuous implementation, so that entities persist over time.

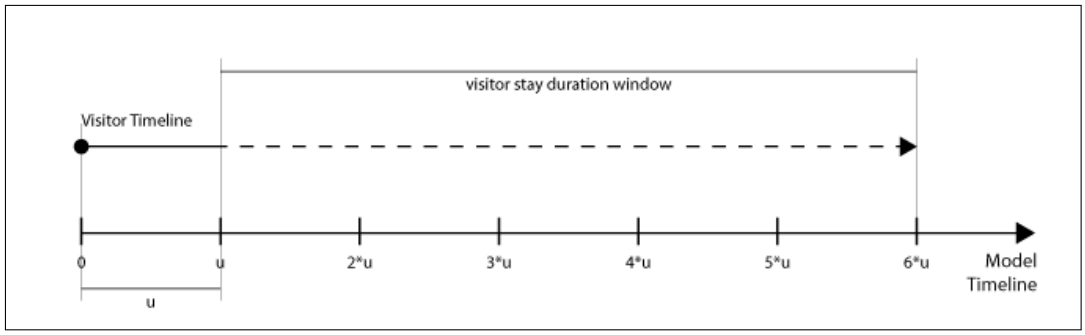


Figure 5.7: Visitor Timeline within the Model Timeline

5.3.2 Applicability of the Agent-Based Modelling Paradigm

In summarizing the previous section, three aspects have been identified to be of primary importance regarding the SDM, and need to be addressed in the implemen-

tation. These are the accurate representation of public space user activities further identified as moving and stationary activities, interactions between different users/visitors of public space, and a persistent/continuous implementation of the SDM. Considering these as the requirements for the development of the sub-model, this section will discuss the ABM paradigm as a framework suitable for the implementation of the model.

Starting with the first requirement, that of capturing user activities: Human movement activity in public space at this scale is encompassed almost in its entirety in pedestrian movement. Numerous examples in literature have been discussed at length elsewhere in this work (*Section 3.3.1: Agent-Based Models of Pedestrian Movement*), in which the specific problem of human pedestrian and crowd movement has been studied using the ABM paradigm, and the paradigm has been found to be suitable for the task. Regarding stationary activities, according to Bonabeau (2002) ABMs provide the following two benefits: They are most natural in describing a system composed of 'behavioural' entities, and they are flexible. Considering these two benefits in conjunction with the existing body of work on modelling pedestrian movement, it is argued here that stationary activities as realized through human behaviour constitute a system composed of behavioural entities, and thus the ABM paradigm is suitable for modelling such a system, and furthermore they can be implemented as an extension of existing pedestrian movement ABMs, due to the paradigm's flexibility.

Considering the second requirement, that of user interaction in a disaggregated model: Given that such a model focusses at the micro-scale, it is assumed that an Individual-Based Model (IBM) would offer a suitable approach for development. Furthermore, considering the continuous nature of the spaces of interest, the heterogeneity of activities, and the focus on entity interaction, the applicability of other IBMs such as Cellular Automata (CA) and Microsimulation Models (MSMs) is questionable, as CA function on a fairly rigid spatial configuration and generally do not differentiate between environment and entity, while MSMs focus more on

individual entity behaviour in isolation, rather than in interaction with other entities. Therefore, ABMs seem to offer the most suitable framework for developing a Spatial Disaggregation Model (SDM) of Public Space Use (PSU).

Regarding the requirement for model persistence: ABMs can be compared to the Object-Oriented Programming (OOP) paradigm often found in modern programming languages, and indeed similarities between the two frameworks have been highlighted (Crooks et al., 2018). In OOP, methods and procedures are considered as standalone *objects*, that can manipulate their own properties, and interact with other objects. They exist within the overall scope of the program until they are destroyed, or the program is terminated, and until that point are able to interact with other objects within the program scope (Kay, 1993). In a similar fashion, in ABM, agents can exhibit persistence over a long period of time, and are able to interact with other agents in the simulation, as long as they are within scope. Therefore, a persistent ABM can be implemented in such a way, so that the simulation runs for an extended period of time, and agents are introduced, remain within the simulation, and can interact with all other agent entities, regardless of when other agents were introduced[3].

Given the arguments presented here then, the SDM will be developed using the ABM paradigm. A detailed description of the public space activity model will be presented at length in *Chapter 7: Modelling Spatial Behaviour*, where the proposed ABM of PSU is presented using the Overview, Design concepts, and Details (ODD) protocol (Grimm et al., 2010).

5.4 Model Implementation

In the previous section it was established that the overall model would be developed using the ABM paradigm. This section will discuss specific development platforms, and will identify the most appropriate environment for the implementation of the

[3]or 'generated', following the general concept of 'generations' in ABM

ABM. For the platform selection process, three requirements have been identified which the development platform should fulfill. First, it should be capable of supporting the development of an ABM. In broad terms, this narrows the selection to any platform that is explicitly designed for ABM and can therefore potentially support the development of the model in this thesis, or a platform that supports a (preferably widely used) programming language that implements a main event loop in order to implement the dynamic model (or both, an environment for ABM development that has a programming interface). Given the specific characteristics and novelty of the proposed model in this thesis, it was decided that the best option would be to develop the model from the beginning using a programming language. Second, the chosen programming language should support Object-Oriented Programming (OOP), as the similarities between ABM and OOP have been discussed previously, and the selected environment should take advantage of this. Third, it was decided that the ABM of PSU would be developed in a three-dimensional environment, and therefore the chosen platform should be capable of supporting a 3D ABM, as well as provide a convenient environment to edit the 3D geometries. The reason for this decision is as follows: At the architectural/human scale, perception of the environment relies on the third dimension (height), and spatial activity observed at this scale is influenced by elements that are inherently three-dimensional[4]. Therefore, a model that aims to simulate user activity at this scale within feature-rich environments should take into account the third dimension, and therefore a 3D ABM would be necessary.

Given the requirements discussed above, two main platform categories were identified. The first category is dedicated *Agent-Based Simulation Platforms*, environments built specifically for the development of ABMs, with the added requirement of being capable of supporting 3D models. A review of ABM platforms (Railsback et al., 2006, Crooks et al., 2018) identified four widely used platforms: NetL-

[4]Including for example walls and building facades that define open public space, as well as features within public spaces, such as stairs, bridges, underpasses, elevations platforms, ledges, etc.

ogo[5], SWARM[6], MASON[7], and Repast[8], and furthermore the JAVA programming language was identified as the language most commonly used in ABM platforms (Nikolai and Madey, 2009). Based on comments from the reviews mentioned here, Repast was chosen as the best candidate from all dedicated ABM platforms.

The second category is 3D modelling environments that support a programming language with a main event loop. The characteristics of this second category are found in modern *Game Development Platforms*, which are tools dedicated to the development of computer games, and thus support 3D geometry (for developing game visuals) and event loop-based programming languages (for implementing game logic). The three most widely used platforms were identified to be Unity3D[9], Unreal Engine[10], and GameMaker Studio 2[11]. Of the three, GameMaker is oriented towards 2D games and was discarded as a potential option. Unreal Engine supports programming using C++ as well as 'Blueprints', a node-based visual scripting interface, while Unity supports the C# programming language and the extended .NET Framework. Of the two, it was decided that Unity would be the best candidate, due to its level of maturity compared to Unreal Engine.

Comparing the two options based on the original criteria, both Repast and Unity are found capable of developing an ABM, although Repast is a dedicated ABM platform, and therefore development in Repast might be more efficient. Repast uses the JAVA programming language, while Unity uses C#, both OOP languages. Models developed in Repast are generally found to be in 2D with the platform supporting 2.5D visualisation and potentially fully 3D models, but would need to be implemented through code along with tools for importing and manipulating 3D geometry, while Unity has native support for 3D mesh geometry and presents a 3D cartesian environment by default. Therefore a conscious decision was made to de-

[5]https://ccl.northwestern.edu/netlogo/
[6]http://www.swarm.org
[7]https://cs.gmu.edu/ eclab/projects/mason/
[8]https://repast.github.io/
[9]https://unity3d.com/
[10]https://www.unrealengine.com
[11]https://www.yoyogames.com/gamemaker

velop the model in Unity, as it was estimated that manipulating 3D geometry would play a somewhat significant part in the model, and therefore a set of 3D editing tools was necessary. Some additional, secondary comparisons between the two: Repast does not have a default model viewer, although it is rather straightforward to implement a basic top-down view, while Unity supports virtual cameras for rendering 3D scenes. Repast has high performance libraries for running the models in computing clusters, while Unity can potentially support some form of distributed computing if implemented; however for the purposes of this work, high performance was not necessary. Finally, as Unity is not a dedicated ABM platform, it has the drawback of not having extensive libraries specifically for ABM development; however some tools from game development can be used for ABM development, such as wayfinding libraries.

5.5 Summary: Building a Real-Time Agent-Based Model of Public Space Activity

This chapter presented an outline and general overview of the general Real-Time Agent-Based Model of Public Space Activity. Initial considerations regarding its temporal nature were discussed, and the balance between near-future and near-past events was discussed as an approach to real-time modelling. Following that, the different sub-models were discussed, specifically the aggregate forecast sub-model, and the spatial disaggregation model, first as inter-connected components in the overall real-time model, and then each one separately in more detail. Two different approaches for the forecast model were discussed and compared, and the most suitable one was identified. Additionally, an extended discussion was offered on specific aspects of the Spatial Disaggregation Model (SDM), which on the one hand established the requirements at a conceptual level, and on the other highlighted a highly suitable modelling paradigm, as identified in the ABM paradigm. Finally, Unity was identified as the development platform, and its features were briefly discussed. In following chapters, the data capturing and analysis methodologies used

in this work will be presented (*Chapter 6: Data Collection and Analysis*), specifics of the SDM will be discussed in more detail (*Chapter 7: Modelling Spatial Behaviour*), the overall model application will be presented through two case studies (*Chapter 8: Case Study 1 - Hyde Park*, *Chapter 9: Case Study 2 - Queen Elizabeth Olympic Park*), and a discussion on results and final model outcomes will be offered (*Chapter 10: Discussion on Case Studies*).

Chapter 6

Data Collection and Analysis

In this chapter, the methodologies developed for the capture and analysis of data are presented, from observational site surveys, to data collection using Application Programming Interfaces (APIs), to data mining of geographic data from social media platforms. The different approaches developed in this work will be discussed in depth, covering the data sources themselves, techniques implemented for capturing the data, both automated, as well as manual, methods employed in manipulating and cleaning up the resulting datasets, initial data analysis and preparation, as well as initial results, findings, and observations regarding the datasets collected. The use of data in the development of real-time agent-based simulations will not be discussed in this chapter, but rather in the corresponding chapters discussing the two case studies undertaken in this work.

Given the scope of this work, of simulating public space activity in real-time, multiple datasets were considered, for different purposes. First of all, major focus was placed on datasets relating to individual human activity on the ground. In order to capture such activity, this work collected data from micro-blogging social media platforms, specifically Twitter and Instagram, which allowed for capturing individual users' activity as it was being published (in real-time). In addition to social media platforms, data on user activity was available from mobile device connectivity records via wireless network access devices, installed throughout the area of one

of the case studies (Case Study 2: Queen Elizabeth Olympic Park (CS2:QEOP)). This dataset includes detailed records of mobile devices, which are often carried on the person, connected to the wireless network of the park. The above datasets were considered as a proxy for actual human activity, able to be collected in real-time. Additional data of user activity was collected via site surveys, in which counts of visitor activity were recorded manually in situ. This data was considered as ground truth data, reflecting actual events, and was used mainly for evaluating and calibrating the models developed in this work.

In addition to user activity data, a range of other datasets were considered, capturing events that were deemed to potentially have an effect on human activity. One such dataset is real-time weather and environmental data. Given the nature of the areas investigated here, being public open spaces hosting leisure activities, weather conditions were considered to have a major effect on the number and type of activities taking place in a space at any point. Furthermore, data was collected, for a limited period of time, on planned events that took place, which would be expected to draw additional crowds on top of normal conditions. These were captured through the Facebook social network platform, which allows users to create events and invite other users to them. Finally, transport data was considered, as a proxy for people arriving in the general area of interest, to be used as an indicator of potential visitors.

Overall, data sources considered in this work, along with their intended uses, are as follows: Online platforms Twitter and Instagram were used to capture geolocated Social Media (SocM) events, with the aim of calibrating and incorporating them into the forecast sub-model (as described in section 5.2). For planned events, Facebook events were gathered, with the purpose of incorporating into the forecast sub-model as well. Weather and climate conditions were retrieved using the Dark Sky platform (formerly forecast.io), aiming to inform the forecast sub-model. Transport and passenger data was retrieved via the Transport for London (TfL) Application Programming Interface (API), in an attempt to develop the alternative visitor supply forecast sub-model (as described in subsection 5.2.2). Wifi connectivity data was provided

by the London Legacy Development Corporation (LLDC), and was used to validate overall model performance, as well as for cross-validation of user activity survey data. Finally, ground truth data of user activity was captured though site surveys, and was used to calibrate the Spatial Disaggregation Model (SDM) (discussed in *Chapter 7: Modelling Spatial Behaviour*). This information is summarized in Table 6.1.

Dataset	**Data Source**	**Purpose**	**Dataset Accessibility**
Social Media Posts	Twitter	forecasting activity	Publicly Available
Social Media Posts	Instagram	forecasting activity	Publicly Available
Planned Events	Facebook	forecasting activity	Publicly Available
Weather Conditions	forecast.io	forecasting activity	Publicly Available
Wifi Connectivity	LLDC	validation	Restricted Access
Transport & Passenger Data	TfL	forecasting activity	Publicly Available
Visitor Spatial Activity	Own Site Surveys	validation & calibration	Publicly Available by Site Visit

Table 6.1: Datasets Used

The rest of the chapter will discuss in detail the different methodologies used to capture, manipulate, and make use of the different datasets presented here. The datasets will be presented by method of acquisition and data source. As such, the four sections will cover online and social media data, WiFi connectivity sensor data, manual site surveys, and transport data.

6.1 Online Data - Real-Time Data

This section discusses the methodologies developed and used for retrieving data from online sources (remote sensing). Data sources include social media platforms, environmental datasets, among others. These datasets' characteristic is their streaming, real-time nature, as it has been defined in previous chapters. Hence, datasets and data points discussed here are available at the moment of capture, and refer to

an ongoing phenomenon.

6.1.1 Social Media Data

Social Media (SocM) data in this work was collected from 3 platforms, Twitter, Instagram, and Facebook. Regarding Twitter and Instagram, focus was placed on individual users' geolocated posts, whereas Facebook was used to retrieve planned events taking place in the areas of interest. In broad terms, the rationale behind this data collection is that individual Twitter and Instagram posts would function as a real-time proxy for current visitor activity in the areas of interest, while Facebook events would offer an indicator and partially account for observed increased activity. Two points of discussion need to be introduced here: first, the real-time nature of online SocM, and second, potential bias or other issues that arise through the use of online SocM data.

On the first point, and the real-time nature of SocM: The nature of micro-blogging platforms themselves encourages users to publish updates in real-time, for quick consumption. It is this characteristic that has attracted interest from researchers, who have investigated the dissemination of information in real-time through such platforms, with particular interest on the use of Twitter in emergency response and disaster detection (Mills et al., 2009, Sakaki et al., 2010, Cassa et al., 2013, Jongman et al., 2015, Avvenuti et al., 2016), but also in urban real-time traffic monitoring (D'Andrea et al., 2015, Kokkinogenis et al., 2015). It is generally agreed then that events published through micro-blogging platforms are of a real-time nature, even if their veracity and predictive capabilities are under study.

On the second point, that of data source bias: It is well documented that there exists a demographic representation bias in online SocM, with populations using online platforms at different degrees, varying by age, gender, and education, among others (Greenwood et al., 2016). Additionally, content analysis of SocM datasets tends to produce skewed results, when compared with offline/traditional surveys (Miller et al., 2015, Cohen and Ruths, 2013). This work avoids any issues that

might arise regarding content, as it does not undertake any content analysis, but rather focusses on SocM dataset metadata to capture relevant information. Specifically, this work approaches Twitter and Instagram datasets as Ambient Geospatial Information (AGI) (similar to Stefanidis et al., 2011), capturing a post's location and timestamp, and discarding all other information.

6.1.1.1 Sources and Capture Methods

Data collection of SocM posts was performed using scripts written in the Python programming language. Scripts were scheduled to run on midnight every day, and collected relevant posts published in the last 24 hours. For Twitter data, the *tweepy*[1] python library was used to access the Twitter Search API[2], for Instagram data a url request was used to access the Instagram API media endpoint[3], while Facebook's API was accessed using the *Facebook SDK for Python* library[4]. The Twitter and Instagram search queries included an empty string for relevant search terms, so that all results would be returned.

Two spatial filtering methods were implemented for Twitter and Instagram data, to return results originating from within the areas of interest. First, a search radius was included in the search terms, so that events were returned only within a certain distance from the center of the area, essentially applying a broad filter. Additionally, by passing this spatial parameter in the query terms, the API automatically filters out any results that lack geolocation, in both Twitter and Instagram API. A second spatial filter was used for finer detail, in order to remove results that fell outside the detailed boundary of the area. For this, a *Point in Polygon* function was implemented, as described in Appendix A.2.

An additional filter was implemented for Twitter and Instagram data, to remove multiple consecutive posts from the same user. The 30 minute mark was used as

[1] http://tweepy.readthedocs.io/
[2] https://dev.twitter.com/rest/public/search
[3] https://www.instagram.com/developer/endpoints/media/
[4] https://facebook-sdk.readthedocs.io/

the cutoff period, so that after a post was captured from user A, any additional posts in the next 30 minutes from the same user would be discarded. This was done to account for differences in usage patterns between different people, as for example oftentimes Twitter users might want to exceed the 140 character limit, by posting multiple tweets in rapid succession. Since the collection algorithms developed here do not store user information, these posts would appear as coming from different users, and thus artificially increase the number of visitors recorded.

Twitter and Instagram data was then stored as CSV files, containing all posts of the past 24 hours. Daily data was stored as a list of individual posts, in chronological order, with each row containing a unique id, the timestamp of publication time, a pair of latitude and longitude coordinates, and a source platform identifier. Overall collection started on September 14th 2015 for the first case study in Hyde Park, and on January 28th 2016 for the second case study in Queen Elizabeth Olympic Park. Although data collection continued for two years, due to changes in Instagram's API, on May 31st 2016 Instagram data collection was terminated for both case studies.

6.1.1.2 Preliminary Data Analysis and Cleanup

Some initial characteristics and general properties of the dataset can be seen by looking at the raw data overview, with social media posts shown in daily totals for a duration of 560 days for Case Study 1: Hyde Park (CS1:HyP) in Figure 6.1, and 424 days for CS2:QEOP in Figure 6.2. Values vary greatly, from the low hundreds to almost 5000 for Hyde Park (HyP). Zero values indicate collection failure, days where the automated collector scripts were not executed properly, and thus no data was captured for that day. Two important things should be noted here: First, it becomes immediately apparent from Figure 6.1 that Instagram data is much larger, totalling approximately 10 times more daily SocM posts. Second, the stop date for Instagram data collection (31st May 2016) is apparent, with a large value drop.

Given the sharp decline in values due to one source becoming inaccessible, and due

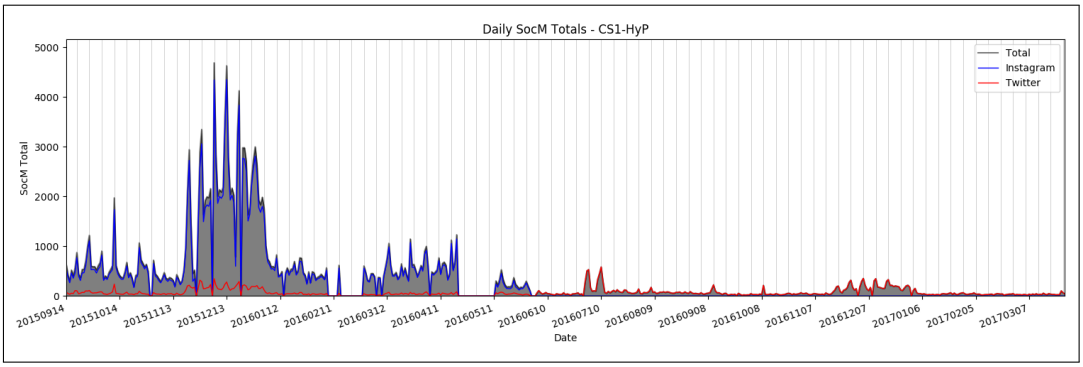


Figure 6.1: SocM Time Series - CS1-HyP

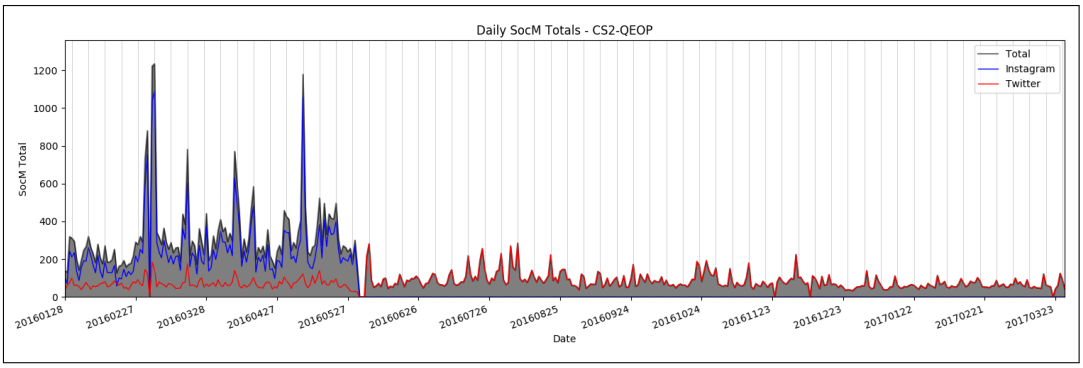


Figure 6.2: SocM Time Series - CS2-QEOP

to Twitter's overall small daily sample, the 31st of May 2016 will be considered as the end date for data collection, and all subsequent analysis of SocM datasets will not include future dates after the end date. Moving forward, regarding weekly fluctuations in daily totals: Vertical lines in Figures 6.1, 6.2 mark Sundays, which given the nature of the two spaces (parks) are expected to have increased visitors, and as can be seen in the figures, spikes in values broadly correspond to Sundays. By further plotting daily totals by day type (Figures 6.3, 6.4), it is apparent that weekends in general seem to attract larger crowds.

6.1.2 Weather Data

Weather and climate data was collected for a significant duration during this work, starting on September 14th 2015, coinciding with the beginning of the collection period for social media data on the first case study. Weather and climate data was collected under the rationale that the areas under examination are public open spaces

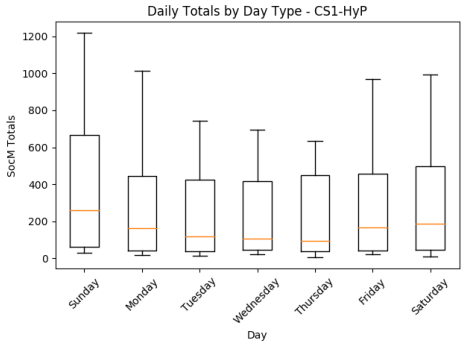


Figure 6.3: Daily Totals by Day Type - CS1-HyP

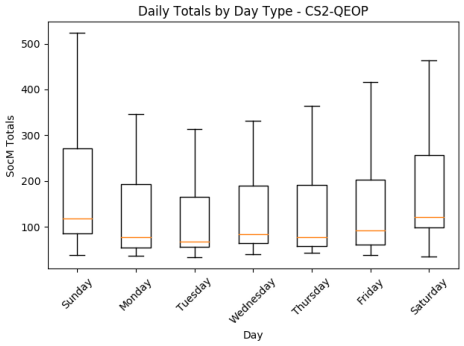


Figure 6.4: Daily Totals by Day Type - CS2-QEOP

hosting largely non-work related activities[5], and therefore the presence and number of such activities would be affected in large part by weather conditions.

The web service *forecast.io*[6] is used to retrieve weather conditions in machine-readable format. This particular service aggregates a range of weather data sources[7], and provides forecasts as well as archived past weather data. Weather data is provided through an API, by passing a pair of coordinates in the request url[8], and will return a set of future forecasts or past weather conditions (depending on requested time). The response is in JSON format with multiple properties. Specifically, the response includes weather conditions at a daily, hourly, and potentially minutely (if requesting a near-future forecast) resolution.

In the context of this work, a set of parameters that could be identified as broad weather descriptors was chosen. The main arguments for choosing a parameter were the following: The parameter should be reliably returned, or its absence directly relating to a value (e.g. the 'cloud cover' parameter refers to the percentage of sky occluded by clouds; its absence signifies clear skies, thus a value of 0 can be inferred). The parameter should conceivably and fairly directly affect open space activity (e.g. parameters such as 'visibility' or 'windBearing' were not included, as

[5] In the sense that users engaging in activities in these spaces are not required to be at that location at any point in time, but rather choose to be present

[6] https://darksky.net/dev/docs

[7] https://darksky.net/dev/docs/sources

[8] https://darksky.net/dev/docs/forecast

Parameter	Abbreviation	Unit
Hour	hr	#
Temperature	temp	C°
Minimum Daily Temperature	maxTemp	C°
Maximum Daily Temperature	minTemp	C°
Precipitation Probability	precP	percentage (0-1 range)
Precipitation Intensity	precInt	inch/hour
Cloud Coverage	cCov	percentage of sky occluded by clouds (0-1 range)
Wind Speed	wndSpd	mph

Table 6.2: Weather Parameters

visibility distance or wind direction would have minor, if any, effects on a typical park activity such as a walk; however, the 'windSpeed' parameter was included, as strong winds would potentially deter park visitors). The full list of captured weather parameters is: Temperature (also min and max daily temperature, for daily resolutions), Precipitation Probability, Precipitation Intensity, Cloud Coverage, and Wind Speed. They are listed in Table 6.2, along with their units, where applicable.

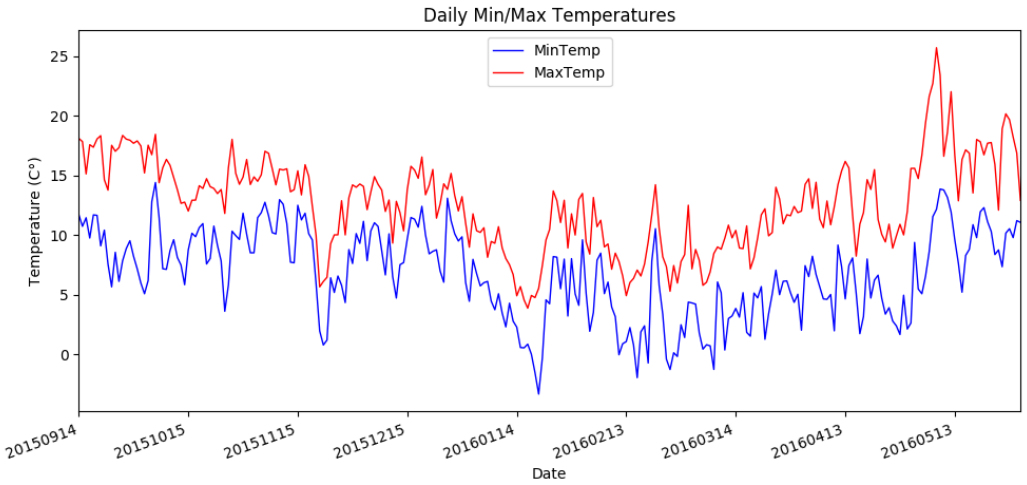


Figure 6.5: Daily Min & Max temperatures

An overview of weather data collected is presented in Figures 6.5, 6.6, 6.7, covering Temperature, Precipitation, Cloud Coverage, and Wind Speed at a daily resolution, for the period between 14/09/2015 - 31/05/2016. The end date coincides with the termination of service of one of the two SocM data sources, and signifies the period for which data sets were considered to be available in full.

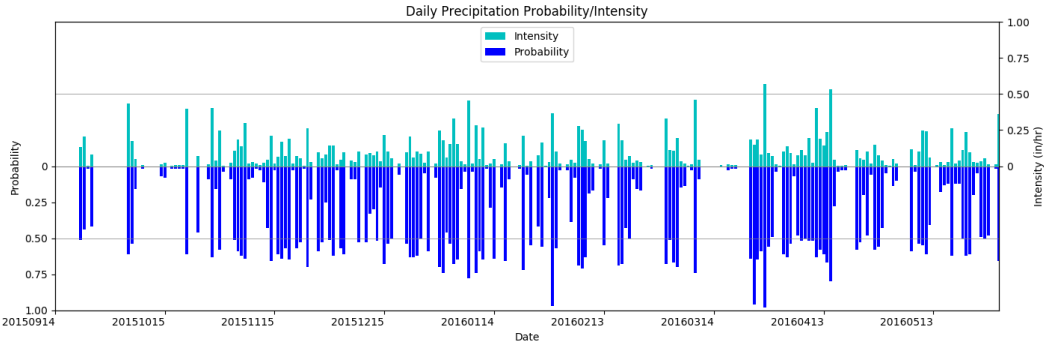


Figure 6.6: Daily Precipitation

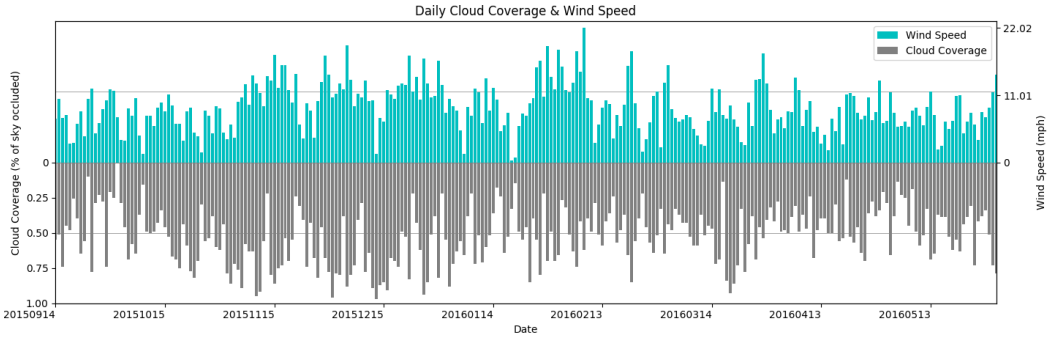


Figure 6.7: Daily Cloud Coverage & Wind Speed

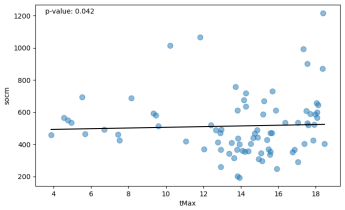
6.1.3 Real Time Datasets - Correlations

The aim of this analysis is to investigate the effect of environmental and temporal characteristics on public space use (measured as Social Media (SocM) posts) during normal conditions. In this context, days with planned events are considered known outliers, with artificially high values. As such, days with planned events, along with zero value days (failed recordings), will not be considered for the rest of this analysis, as these records would introduce a strong bias. Even having removed known outliers, increased activity on Sundays is further evident when comparing SocM by day (Figures 6.3, 6.4). Most SocM are recorded during Sundays, averaging 750 daily total, with values falling sharply on the next days, and picking up again on Saturdays.

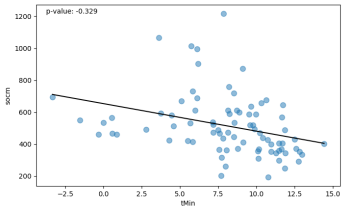
Daily Aggregate This section will be looking at the effect that climate and temporal characteristics have on recorded social media posts, first at the daily aggregate level,

and later at an hourly level.

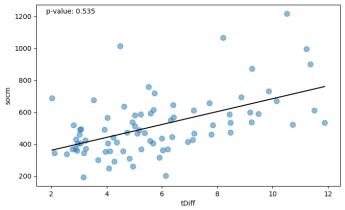
At a daily aggregate level, initial assumptions focused on temperature being the main driver of park visitor activity (and thus social media activity), stating that days with higher temperatures would attract higher visitor numbers. This turned out to be a false hypothesis, as can be seen on Figures 6.8a, 6.8b, showing daily SocM levels against daily maximum and minimum recorded temperatures.



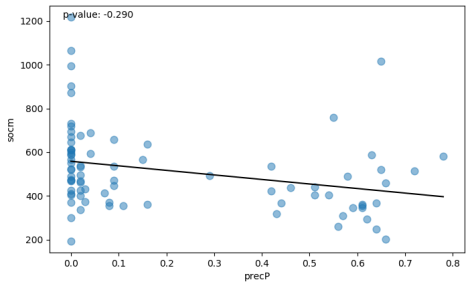
(a) SocM - Max Temperature



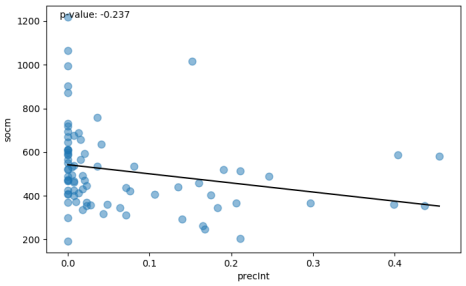
(b) SocM - Min Temperature



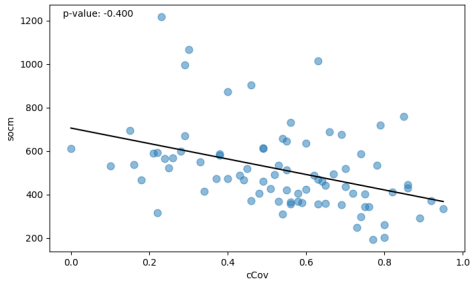
(c) SocM - Temperature Difference



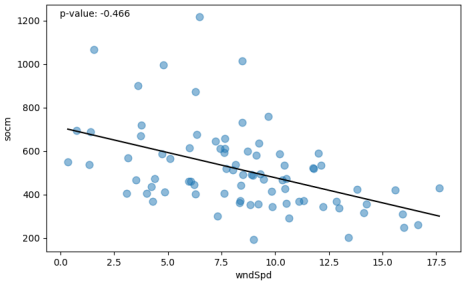
(d) SocM - Precipitation Probability



(e) SocM - Precipitation Intensity



(f) SocM - Cloud Coverage



(g) SocM - Wind Speed

Figure 6.8: SocM - Weather Daily Correlation

It is evident from the graph that no correlation exists between maximum temperatures and SocM, at least for the time range in question, while daily minimum temperatures exhibit some degree of negative correlation with SocM. It is interesting to

note though that temperature difference between maximum and minimum recorded daily temperatures provides the best fit of the three variables from a statistical point of view, with a positive correlation Figure 6.8c. However, as temperature difference does not directly relate to an attribute that could explain this behaviour, analysis turns to other climate characteristics, more specifically cloud coverage, wind speed, and precipitation probability and intensity, which should at the same time affect SocM as well as temperatures. These characteristics are known to affect ground temperatures (Easterling et al., 1997), and can furthermore be considered as creating unfavourable conditions for park visitors, thus reducing visitor numbers.

Cloud coverage exhibits a negative correlation with SocM, with a strong (for the dataset) fit, as seen on Figure 6.8f. Similar results are displayed when comparing SocM against wind speed Figure 6.8g, indicating that unfavourable weather conditions have a negative impact on park usage, as would be expected. SocM and precipitation exhibit a similar relationship, although not linearly correlated. As seen on Figure 6.8d, for precipitation values greater than 0 (chance of precipitation), SocM values average at about 400 daily total posts, providing a potential baseline of park activity regardless of weather conditions, possibly indicating restaurant visitors and less weather-dependent activities, such as exercise activities.

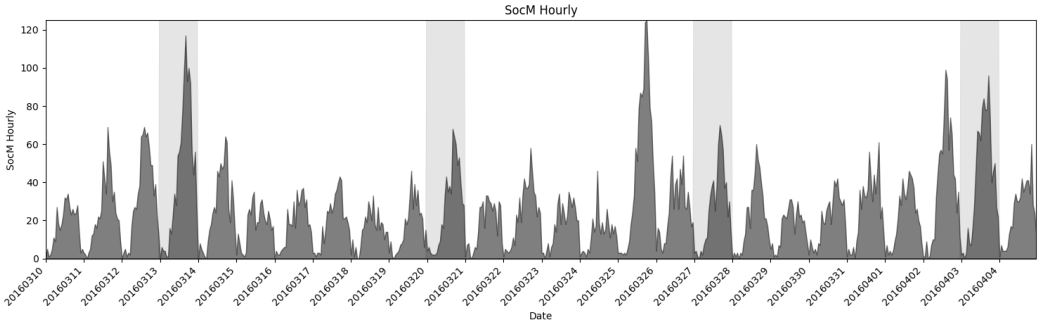
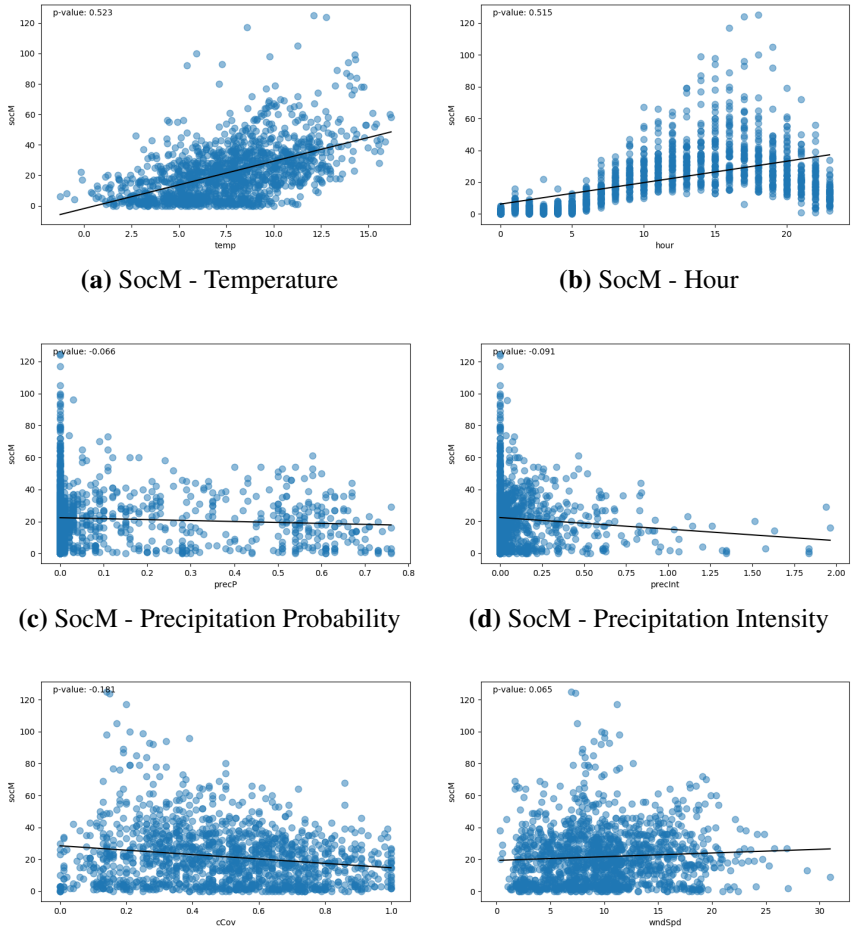


Figure 6.9: SocM Hourly

Hourly Aggregate Analysis at a daily resolution identified some weather characteristics as broad drivers of park visitor activity, as shown previously. In this next section, activity will be investigated at an hourly temporal resolution, in order to capture the relationship between SocM and weather/temporal characteristics in more

detail. SocM data exhibits fairly consistent periodic characteristics, following the day/night cycle, as can be seen in Figure 6.9. Looking at hourly SocM totals against hourly temperature, as shown in Figure 6.10a, it again becomes clear that on the whole, there exists some correlation between temperature and park visitor activity, however climate conditions do not appear to be the sole driving factor.



(a) SocM - Temperature

(b) SocM - Hour

(c) SocM - Precipitation Probability

(d) SocM - Precipitation Intensity

(e) SocM - Cloud Coverage

(f) SocM - Wind Speed

Figure 6.10: SocM - Weather Hourly Correlation

Results of minimal correlation are exhibited when looking at other weather characteristics at an hourly temporal scale, such as cloud coverage (6.10e) or wind speed (6.10f). Data points in these cases are scattered with no discernible patterns, with the exception of precipitation, where, as expected, SocM values are at their constant lowest (approximately 20 per hour) when any rainfall is recorded. Of course, this behaviour of no relationship at hourly levels is expected. Given the temporal scale

of 15 minutes, variation in SocM is caused more by hour of day and daily activity cycles, than any other climate characteristic. Following this reasoning, a very discernible pattern is exhibited when looking at SocM by hour of the day, as seen in Figure 6.10b.

Hourly SocM values are at their lowest during early morning hours, between midnight and 5 am, with valley values at 2 am. Activity starts to pick up at 6 am, and rises steadily until a peak is reached at 3 pm. After this hour, values decrease again steadily into the night, until they are at their lowest at 2 am again. This oscillation in SocM values can be modelled using a 4th degree polynomial, in the form of $y = ax^4 + bx^3 + cx^2 + dx + e$, with $a = 0.001, b = -0.065, c = 1.15, d = -3.8, e = 4.87$, which when fitted to the data points, results in a coefficient of determination of 0.47 (Figure 6.11).

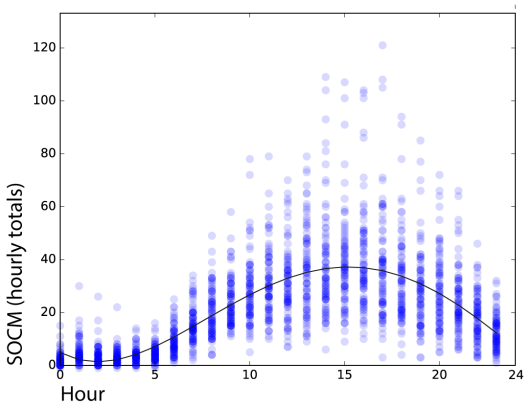


Figure 6.11: SOCM - Hour-of-Day: polynomial fit

6.2 Sensor Data - WiFi

This section discusses data relating to device connections over near-distance wireless networks, capturing mainly mobile devices carried on the person, and thus is used as a proxy for activity in the area. This approach was deployed at the Queen Elizabeth Olympic Park, and it involved the deployment of a large number of network access points (approximately 65 devices) by the London Legacy Development Corporation (LLDC) throughout the park, which record the number of devices cur-

rently connected to them. The locations of the access points are known, and this dataset can therefore be used to infer activity on the ground. Data from this source was available for dates and times of site surveys as well, which allowed for the combinatorial analysis and cross-validation of both datasets.

Detailed WiFi connection data is available for the month of March 2016, consisting of anonymized unique individual connections at each access point. Additional information includes device session duration (total duration this device has been connected to the network), currently connected access point per device (and inversely, total current sessions per access point), volume of data received and transmitted, connection and disconnection time per device per access point. Some discrepancies were quickly identified, in devices connected continuously for extended periods of time (more than 24 hours, and at times significantly longer, i.e. several months), and so a filter has been applied to the whole dataset, removing any records with a total continuous duration of more than 6 hours.

An overview of the dataset is presented in Figure 6.12, a time series of daily totals of connected devices for the whole network. Daily volumes stay fairly consistent throughout the period, with one notable peak (day 20) and one significant dip (days 27-29). Further investigation at hourly totals (Figure 6.13) highlights a strong periodic day-night cycle, with values during nights and early morning hours being consistently low. Therefore, any change in daily totals is largely a result of daily activity.

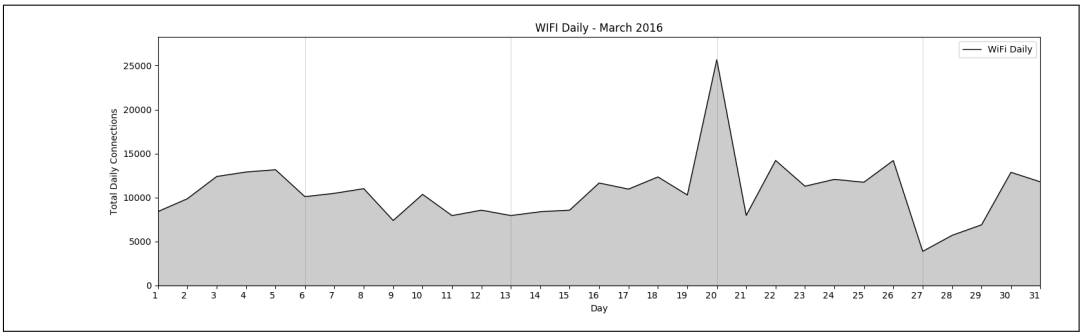


Figure 6.12: Unique Wifi Connections - Daily

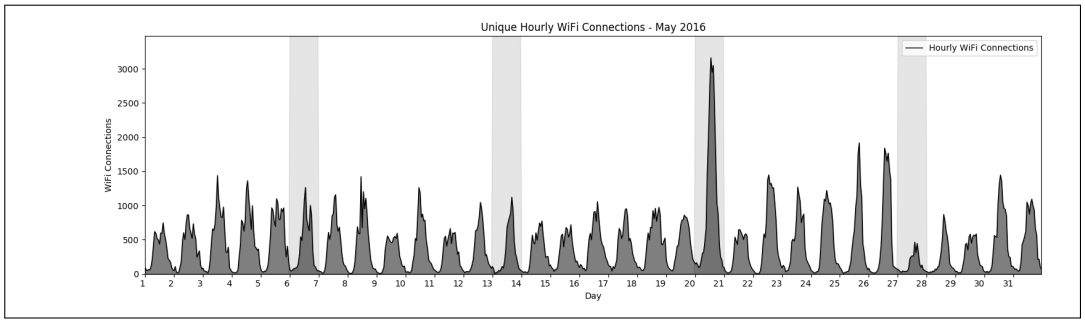


Figure 6.13: Unique Wifi Connections - Hourly

6.3 Site Surveys

For the purposes of evaluating all of the data capturing techniques developed here, a series of site surveys where performed, which captured actual activity on the ground. These function as the ground truth data for all cases, and were performed over multiple days, for both of the case studies undertaken in this work. The aim of these surveys was to record the total number of park visitors at a given moment throughout the area, as well as specific locations of individuals, along with type of activity.

6.3.1 Aim

The aim of the site surveys was to capture ground truth data regarding human activity in the areas of interest. This data was needed first of all to provide context for the rest of this work, and to better frame expectations from the models developed later. Additionally, spatial output from these surveys was used to calibrate the Agent-Based Models (ABMs) developed in this work, discussed in later chapters. Two broad categories of human activity were considered, 'movement' activities and 'stationary' activities.

6.3.2 Methodology

Data was captured using a purpose built application installed on a mobile device[9]. It provides an interface for a series of counters with customizable labels, which when clicked/tapped by the user record a new event of the particular label (Figure 6.14). Additionally, the recording action captures the time of the event in unix time as provided by the Operating System of the device. Furthermore, the recording action captures the geolocation of the device at the time of the event, as provided by the device's GPS sensor. The application is also designed to automatically record an event of default 'GPS' type every 5 seconds, which records the device's/surveyor's current location. This latter functionality is provided for fieldwork over large areas, to provide a track of the surveyor's path throughout the survey. After the survey, the dataset is retrieved as a CSV file, containing every event recorded, sorted in chronological order, essentially a series of space-time events.

Before the site visits, the paths to be taken were carefully planned so as to cover as much of the area as quickly as possible. During the survey, park visitor activity was recorded. The classification included two categories, walking visitors, and sitting visitors. The surveyor strictly followed the path, and recorded all individual park visitor activity that was evident within a range of 100-150 meters. This essentially creates a buffer zone around the path line of 150 meters (Figure 6.15). The surveying application functionality records the device's location when a new event is appended, meaning that all activity is recorded on the surveyor path (Figure 6.16). Any locations outside the area covered by this buffer were not captured. Care was taken to include as many locations as possible, and to capture sharp changes in activity density. Areas that were unrecorded were nonetheless chosen so that they exhibited similar activity to nearby recorded areas (observed during the visit, but not recorded), so that data could be inferred if needed, via a linear interpolation/extrapolation from nearby activity.

[9] *Fieldworker*, developed by researchers Panos Mavros and Katerina Skroumbelou at Centre for Advanced Spatial Analysis (CASA).

Figure 6.14: Fieldworker Site Survey Application

Finally, a compromise was made regarding moving visitors, to only record park users who crossed paths with the surveyor in angles up to 180 degrees. In combination with the surveying distance of 100-150 meters, this means that a cross plane approach was implemented: an imaginary vertical plane centered on the surveyor, spanning 100-150 meters in each direction perpendicular to the forward direction, and facing forward, which when crossed by others, results in the recording of a new event. Essentially this means that any visitors approaching the surveyor from behind were not recorded (Figure 6.17). This compromise was made to avoid potentially double or triple counting moving visitors, who might overtake, then be overtaken by the surveyor while resting, then overtake again, etc. Furthermore, a uniform movement is assumed for all visitors, meaning that park visitors have an equal chance to be moving in any direction in the park. Therefore, by only recording movement

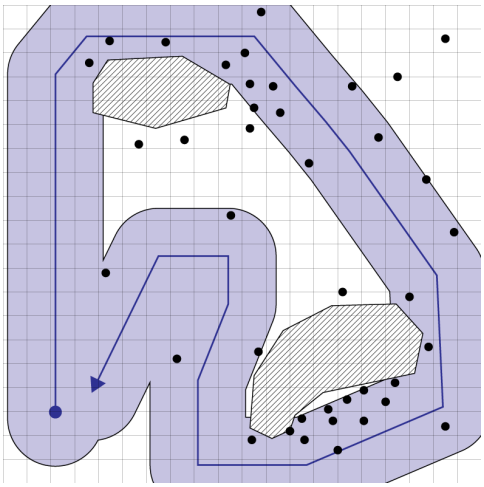


Figure 6.15: Site Surveying Overview

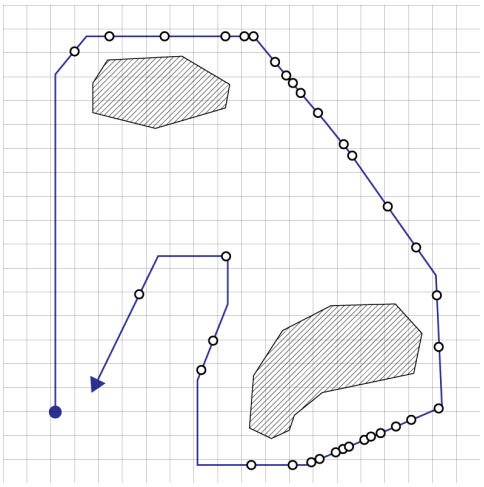


Figure 6.16: Site Survey Result

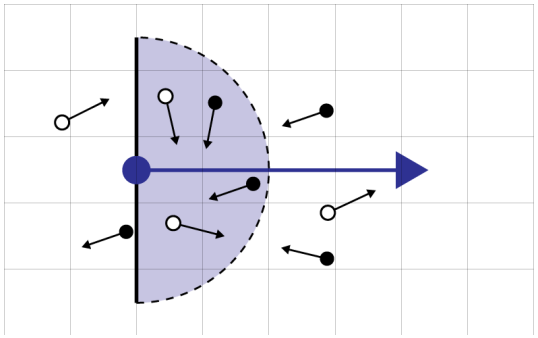
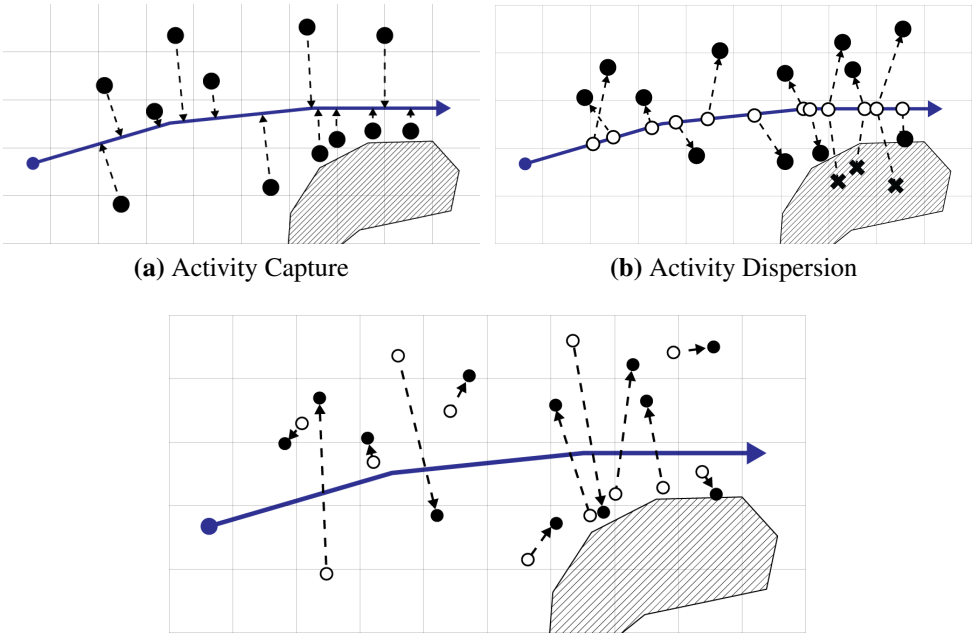


Figure 6.17: Surveyor Cross Plane Capture: Black dots mark pedestrians that were or will be recorded, white dots mark pedestrians that will not be captured, given steady trajectories

happening at angles up to 180 degrees, it is assumed that half of the moving visitors are recorded. Finally, the surveyor walking speed was at a quick pace, overtaken only by park visitors that were at a jogging or running pace.

6.3.3 Data Preparation and Cleanup

The data was imported into GIS software to cleanup, prepare, and analyze. The first step was to disperse individual data points from the path line back to their locations in space. The application records the geolocation of the device at the time of a capture event, and so all recorded visitors appear to be on the path line (since that was the location of the device that was used to record the event) (Figure 6.18a). Therefore, for each event, a new random location was calculated, so that: it was



(a) Activity Capture **(b)** Activity Dispersion

(c) Activity Displacement

Figure 6.18: Survey Activity Dispersion. 6.18a: Events are captured as being on the survey path. 6.18b: In post-processing, events are randomly dispersed around the capture location. 6.18c: Activity displacement between actual locations (black) and estimated (white).

within recording distance (100 meters), and was on valid terrain (e.g. not in water, as water activity was not recorded) (Figure 6.18b). New points were drawn at random, assuming a normal distribution around the surveyor location. The new points were then considered as the actual location of the recorded activity for all analysis regards (Figure 6.18c). The script used for the re-dispersion is shown in Appendix A.5.3.

6.4 Transport Data

An additional approach to estimating activity in public spaces was explored, in which publicly available transport data was considered as an indicator. Essentially this approach assumed public space as the direct receptor of outflows from public transport and thus if the number of people arriving in an area were known, the estimation of activity could be further estimated.

6.4.1 Datasets

Some exploratory statistics regarding the transport datasets available are presented here. Available data is split into two main datasets, one covering long-term statistics for the whole network, the other providing sample detailed data for individual stations.

6.4.1.1 Overall Performance

The first dataset, which includes long-term statistics, is published by GLA under a UK Open Government License (retrieved from data.london.gov.uk/dataset/public-transport-journeys-type-transport). It provides the number of journeys on the TfL public transport network, broken down by mode of transport. Data on London Underground and Bus journeys covers the date range from April 1st 2006 - present. It is a rolling dataset, updated monthly, with approximately 2 months of delay between collection and publication. Temporal resolution is at 28-day periods, totaling 13 periods per year, each period normally starting on a Sunday and ending on the Saturday 28 days after. Years change on the 1st of April each year, which produces varying length effects for the first and last period each month, resulting in edge period lengths ranging from 25 to 32 days. Counts are normalized for all operations on this dataset, either to journeys per day or per typical period length (original value / period length * 28), depending on operation (Figure 6.19).

Data presents a seasonal pattern within years, with an overall increasing trend. There is a notable dip in values from year 2008-2009 to 2009-2010. Following that point in time, data suggests a steady linear increase in journeys. For all calculations henceforth, the first 3 years are discarded, 2009-2010 is used as the first period in the time series, with April 1st 2009 being the first day in the time series.

Linear relationship between subsequent years post-2009 is further suggested by looking at the linear regression plots for average daily journeys over time. Plotting the curve for daily period average trends results in $r^2 = 0.592$ (Figure 6.20), while

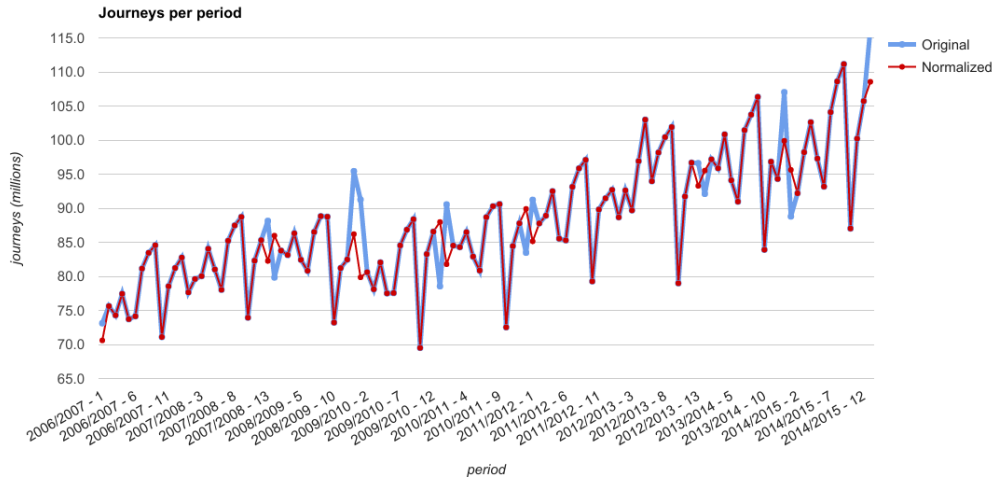


Figure 6.19: Tube Journeys by period: 2006-2015

aggregating at the daily annual average level results in $r^2 = 0.991$ (Figure 6.21), with quite similar intercept and slope, therefore suggesting an acceptable linear overall trend.

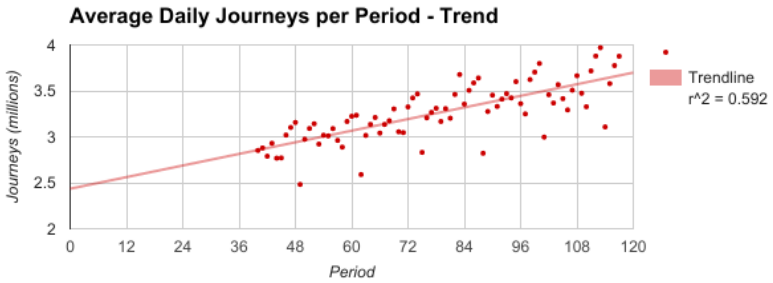


Figure 6.20: Underground Journeys per Period

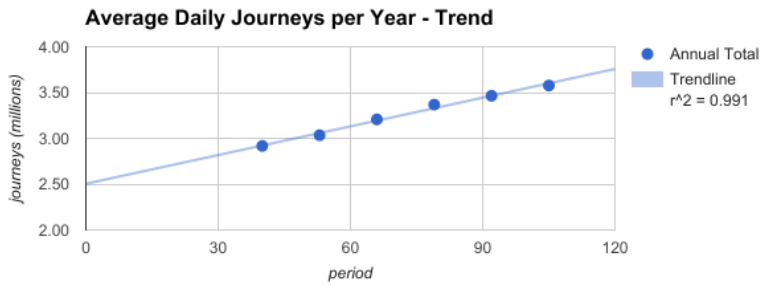


Figure 6.21: Underground Journeys per Year

Regression analysis results hold true for specific period trends as well. A plot of daily averages for 4 periods over the years reveals a similar linear trend, with r^2 values >0.9. An exception here is present in trend lines for periods falling in the

month of August, specifically periods 5 and 6. Including all years post-2009 results in r^2 values of 0.706 and 0.857, due to increased recorded traffic during the Olympic Games hosted in London in 2012. Removing these data points results in r^2 of 0.991 and 0.992 respectively, much closer to overall values. For this reason, further calculations will disregard these two data points.

6.4.1.2 Sample Exits at Stations

The second dataset includes sample detailed data on passenger counts entering and exiting individual stations. It is published by TfL under a UK Open Government License (retrieved from api-portal.tfl.gov.uk). It includes daily passenger counts for each individual tube station, for a typical weekday, Saturday, and Sunday. Temporal resolution is at 15-minute intervals, spanning a full day. Counts are based on an average over five weeks, with the majority of data collected in November-December 2012. This paper focuses on passenger exits, the methods presented here however can be similarly applied to entry data.

Counts at each day start and end at 2:00 am, at which time no trains are travelling, therefore guaranteeing zero passengers and a smooth break between days. For the purposes of this work, data has been reformatted to fit a period starting and ending at midnight, by appending the previous days data from 00:00 - 2:00am to the current day dataset. This results in the introduction of an additional dataset for Monday data, matching Sunday values for 00:00 - 2:00am, and weekday values from 2:00am until midnight.

Exit data for weekdays shows a sharp rise around 9:00am, as is expected from the morning peak time (Figure 6.22). Exit counts remain constant throughout business hours, up until a rise around 6:00pm, for the afternoon peak time, dropping slowly for the evening. Exit peaks alternate expectedly between central and peripheral stations, with central stations showing a sharp spike in the morning and a small bump in the afternoon, and vice versa from the peripheral stations.

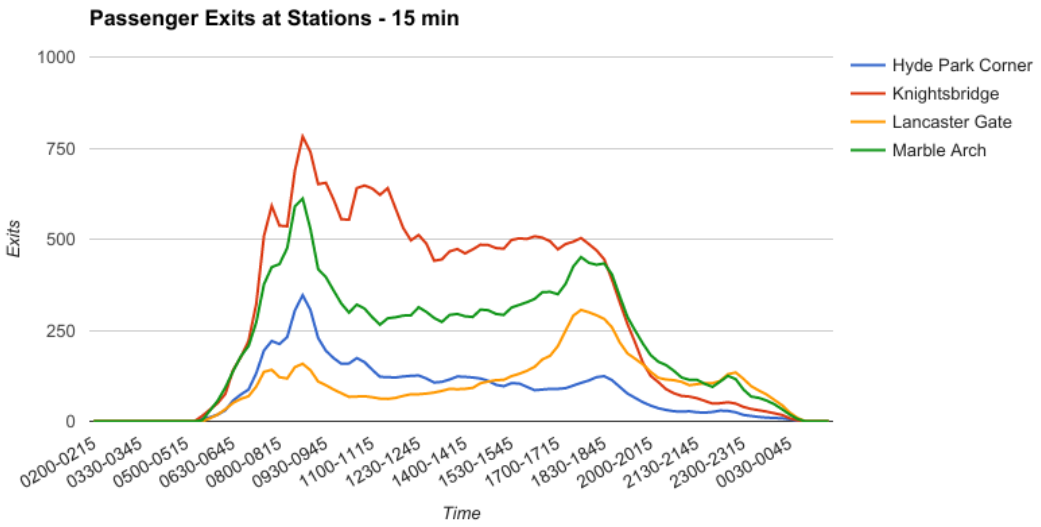


Figure 6.22: Passenger Exits at Stations during Weekdays - 15 min

Saturday exit data shows a steady rise throughout the day from 9:00am until 7:00pm, and a fairly sharp rise during evening hours, with counts staying relatively high until midnight. Sunday data shows a sharp drop from midnight until 2:00am, as is expected from crowds returning from the Saturday night. Throughout the day, exits rise steadily until noon, remaining constant until 8:00pm, after which they drop steadily.

6.4.2 Estimating Real-Time Tube Traffic

This section discusses a method for estimating current use of London Underground network, measured at terminals (tube stations) as passenger exits at a time scale of one minute. The process is carried out in 3 different steps:

1. Interpolation of daily total (disaggregation method)

2. Extrapolation of current (or future) daily total

3. Disaggregation to minute count at station

6.4.2.1 Interpolation of daily total

Disaggregation to daily values is performed using a linear interpolation method (as described in *Appendix A.1*). One of the main problems with the available historic data is the varying length of edge periods in different years. These periods vary in length from 25-31 days, and by using the linear interpolation method, daily disaggregated values are unaffected by this fact. Further notes regarding the application of this method to the specific dataset of Tube journeys are presented here.

Although daily variation throughout the week is evident, for the interpolation and extrapolation stages days are assumed as similar. This produces a value for a typical date for a specific date, which is an unrealistic value, but allows for further calculations without introducing complications at this stage.

Detailed sample data is available for daily variation throughout the week, as well as detailed quarter hour counts per station. This sample data comes from sample counts in November 2012. To be able to work with these detailed data sets, estimated typical daily values are also expressed as a ratio against average November 2012 daily values, acting as a modifier for said dates. This allows for a later application of calculated daily modifiers to detailed sample data, arriving at high temporal resolution estimates.

6.4.2.2 Extrapolation of current and future daily total

This project uses archived journey data to estimate current use via linear extrapolation. Existing data covers the period 2006-present, aggregated at 28-day intervals, with edge periods (first and last in year) of varying length. Using the disaggregation method described previously, this model calculates the values for the date in question (month and day) in previous years, and extrapolates to current date.

Looking at the historic data available, there is a dip in values during the 2009-2010 period, with a steady rise following that. Given the steady rise post-2009, pre-2009 values are not used, and a linear model is fit to remaining dates.

The model calculates values for dates in question in previous years, expressed as ratios against a fixed value, in this case the daily average for period 10 (November-December) of year 2012-2013. A linear curve is fit to these values and the current date value is calculated from that. An exception here is made for dates falling within the period of July 20th and September 14th. In these cases, historic data for the year 2012-2013 are not included, as these periods are a known outlier due to London's hosting of the Olympic Games leading to increased traffic.

Further to the extrapolation, values are calculated as a typical day for that date, ie at this point all days are assumed to hold equal weight, without weekday-weekend variation. This eliminates the issue of a date of interest falling on different weekdays in previous years, skewing the result. Also, by expressing the value as a ratio against November 2012 average, the value acts as a modifier to detailed sample data available from that period. Following the calculation of the daily value for the date in question, by applying the daily modifier to daily totals for the day type (weekday, Saturday, Sunday), the final total daily count estimate is calculated.

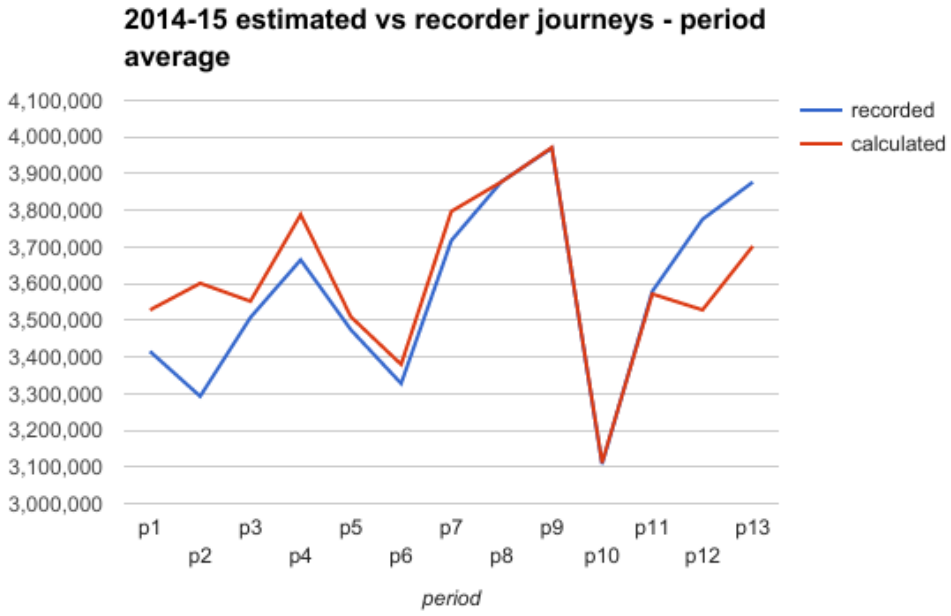


Figure 6.23: Tube Journeys by period: 2014-2015

For validation, values for the year 2014-2015 were calculated using the prediction

method discussed here and subsequently compared to recorded values as published by the authority (Figure 6.23). Values were calculated for each individual period, 13 in total, covering the time from April 1st 2014 to May 31st 2015. Period totals were calculated as the sum of daily totals for all days in period, by estimating each daily total.

Estimated and actual values are fairly consistent, with major discrepancies observed at near-edge periods at both ends (p1, p2, p12, and p13). These discrepancies are attributed to two factors. First, by looking at specific period trends over the years, it is evident that three of the four (p1, p2, p12) show unexpected outlier values.

Second, regarding edge periods (p1 and p13), there is a known issue caused by their variable duration, different from the varying period length issue discussed previously: As mentioned, daily values are calculated as a ratio against a known value of a typical November 2012 day. In this context, typical November 2012 day is calculated as the average of 20 weekday, 4 Saturday, and 4 Sunday totals, essentially assigning mentioned values as weights to different day types. In the case of edge periods of different (not 28 days) lengths, these weights are known to be different than a typical 28-day period, which, when values are converted back from typical to day type totals, result in these discrepancies.

6.4.2.3 Disaggregation to minute counts at station

Detailed passenger exit counts are also available, showing exits per station at quarter hour intervals. This data comes from sample counts during period 10 of year 2012-2013, representing typical days in November. By applying previously calculated daily modifiers to this data, it becomes possible to estimate current quarter-hour exit counts at individual stations (Figure 6.24).

The final step requires the disaggregation of quarter hour totals to minute values. There are a few different approaches, depending on application. A simple approach is that of averaging the total to minutes, each minute having the same flow of exit-

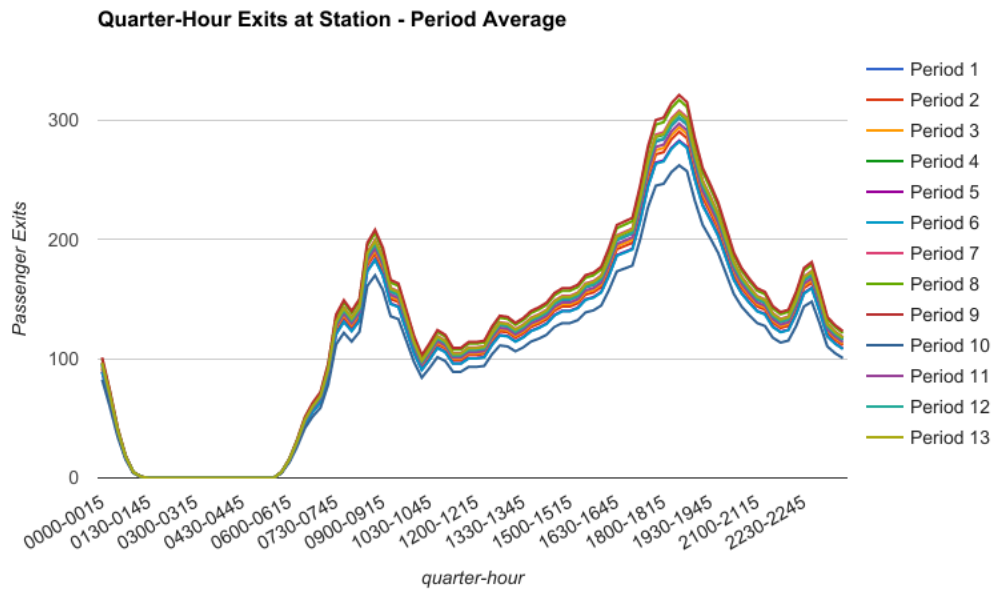


Figure 6.24: Passenger Exits at Station - Period Average

s/minute for the current 15-minute period, with artificial steps introduced between periods. Although this approach is fairly simple and straightforward in regard to disaggregation models, given the already high temporal resolution, it might result in acceptable values, depending on application.

Another approach might be the application of a second round of disaggregation, from 15 minutes to one minute. Since the periods have a fixed 15-minute length, either the linear interpolation method presented earlier, or various other disaggregation methods can be applied, in order to arrive to minute values.

Given the nature and temporal resolution of available data, it being archived and aggregated to approximately monthly periods, this approach presents some notable limitations. Initial steps of disaggregation to daily values and extrapolation to current day values provide an acceptable result, as can be seen when comparing predicted and actual values for the annual period 2014-2015 (Figure 6.23). However, a key limitation is encountered when attempting to disaggregate to quarter-hour values. As the only available dataset at this resolution includes a single weekday, Saturday, and Sunday, any disaggregation at this scale necessarily makes use of

this dataset. This results in unrealistic final values, as individual stations exhibit minimal variation: end result is essentially the quarter-hour sample dataset, scaled slightly according to period (Figure 6.24). Furthermore, even in the case where additional high resolution datasets were available (e.g. smart 'Oyster' card data and individual passenger journeys), the fact that such data is archived continues to pose a limitation within the real-time scope of this work. Additionally, this proof-of-concept approach of estimating current passenger volumes focused on Underground data, which arguably is the mode of transport with the best data coverage. To consider transport infrastructure as drivers of public space activity, additional modes of transport would need to be considered (e.g. buses, taxis), which would potentially require very different data capturing and analysis approaches.Therefore, due to the reasons discussed here, this approach for estimating current public space activity can be ruled out as being suitable for further analysis.

6.5 Summary - Finalized Data Formats

Considering the range of datasets discussed in this chapter, a summary is offered here. This section will briefly highlight each dataset's format and characteristics, and consider their overall properties.

Real-Time Data True real-time datasets identified in this work were social media data (Twitter, Instagram), weather conditions (forecast.io - multiple weather stations), and live public transport arrivals (ultimately not used). Site surveys are non-real-time by definition, as they are non-automated processes. WiFi datasets used in this work are not real-time: although they were collected at fine temporal intervals, they were made available much later (specifically, months), after collation of larger periods. Still, the potential (at least from a technological point of view) exists for WiFi data to be made available in real-time, although that would require further discussion of research ethics, regarding the real-time tracking of individuals. Transport/Passenger data was neither real-time, not actual, as it originated from archived synthesized data, and resulted in estimates. A summary is presented in Table 6.3.

Dataset	Temporal Resolution	Publication Delay
Twitter	Timestamp	None - Instant
Instagram (pre 2016/05/31)	Timestamp	None - Instant
Instagram (post 2016/05/31)	N/A	N/A
Weather	Hourly	None - Instant
WiFi Connectivity Records	Timestamp	Months
TfL Exits at Stations	15 minutes (averaged over multiple weeks)	N/A - Single Publication
TfL Monthly Passenger Volume Statistics	Monthly	Month

Table 6.3: Dataset classification in terms of temporal characteristics

Temporal Characteristics Social media data was considered as repeating/periodic time series, at a daily, hourly, and 15-minute resolutions. Weather conditions were similarly considered. WiFi data was also considered as periodic time series, available at 1-second resolutions, but analyzed at lower resolutions (15-minute and hourly), for the days that data was available. Site surveys were considered as fixed points in space (no temporal continuity).

Spatial Characteristics Social media data was ultimately not considered as spatial data, as its geolocation classification system proved to be too coarse and/or unreliable. Weather conditions were also considered as a-spatial, given that no differences in weather conditions would or could be detected between two different locations in the area of interest. WiFi data was treated as spatial data, at a medium resolution (a data point was considered to be accurate to within 70 meters), although bias was known to exist. Site survey data was considered as reliable in terms of spatial characteristics, with data points accurate to within 70-100 meters.

Chapter 7

Modelling Spatial Behaviour

This chapter discusses the methods used in this thesis to simulate the emergent behaviour of individuals in a virtual spatially-explicit environment. It builds directly on two of the three main fields identified and discussed extensively in this work, specifically Public Space Use (PSU) studies and Agent-Based Models (ABMs), as they have been presented in Chapters 2 and 3 respectively. The ABM paradigm will be used to implement observations of human spatial activity in a simulation environment, with two aims: First, to evaluate and test such findings, through simulation. Secondly, to employ the resulting spatial behaviour ABM in simulations of real-time models of public space activity.

The development process will be discussed in detail, from the codification of observed behaviour, to the development of theoretical models of individual components, to the computational implementation of such behavioural components, to the evaluation of their implementation in synthetic populations via an ABM framework. The ABM framework that will be used to describe the models of PSU developed and implemented in this work is the updated 2nd version of the ODD protocol (Grimm et al., 2010). For the purposes of testing the model during development, a simple representation of a park was created and used that included all of the entities and characteristics required in the model. It does not correspond to any of the two case studies, nor any actual place, and will only be used in this chapter for the presenta-

tion of the ABM developed here.

7.1 Overview

7.1.1 Purpose

The purpose of this model is to realistically simulate spatial public space activity, as generated through the behaviour of individual human users, acting and interacting both between themselves and with the mostly static environment of the simulation. One of the core aims of the model is to produce a framework for perpetual simulations, where a simulation is designed without a predetermined end, but rather is configured to run continuously, with the explicit aim to continuously accurately capture and reflect real-time activity in a space. Therefore this model does not aim to produce an ideal solution to a problem or to find optimal values of parameters, other than the minimization of error between actual and simulated activity.

7.1.2 Entities, State Variables, and Scales

7.1.2.1 Entities

Agents The main entity type in this model is the individual agents, which represent users of simulated space. Agents in the simulation are physically represented by a simple primitive 3D model (Figure 7.1). They are synthetic humans that interact and engage in activities appropriate to the overall environment type. This work focusses on public space, and more specifically parks, as such agents in this model represent park visitors. The agents have behaviours that are classified into two broad categories: Movement (moving to/from a specific location) and Stationary Activities (Agents engage in activities that are considered to be fixed in space - even if the actual activity includes movement eg. sports, the activity takes place within a predetermined area eg. playing field, and thus fixed in space). The agents essentially pick a location at which to engage in an activity or perform an action,

and then move to that location.

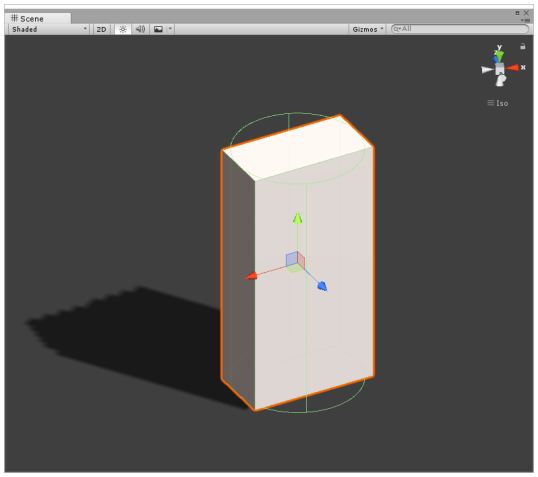


Figure 7.1: Agent Virtual Avatar

Environment The environment within which the agents act and interact is explicitly represented, via a 3-Dimensional virtual model of the actual space (Figure 7.2).

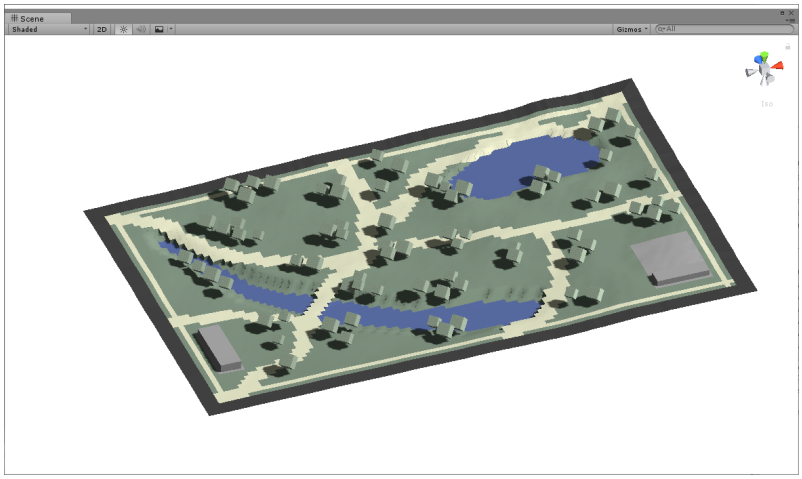


Figure 7.2: Virtual 3D Environment

The environment encompasses all of the static physical elements in the simulation that constitute the environment within which agents act, and that potentially affect agent behaviour. Given the focus on park activity in these simulations, elements in the environment include the different types of terrain (eg. paths, lawns, etc, more in the next section ”State Variables”), trees, features, buildings and points of interest in the park (fountains, restaurants, etc), as well as the gates, designated entrances and exits of the space (Figure 7.3).

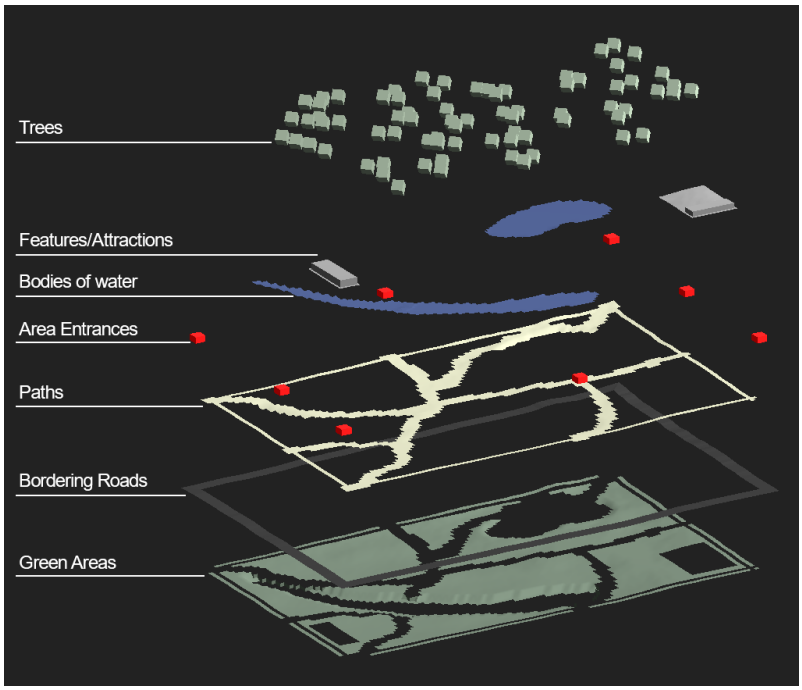


Figure 7.3: Exploded Isometric View of 3D Environment Components

Controller The controller is a singleton high-level entity in the model, tasked with controlling most of the higher-level functions. These include the control of simulation variables such as time, and environmental conditions, such as simulated time of day, day type, weather, etc. More importantly, the number of agents is expected to fluctuate widely during the course of the simulation, and the controller is the entity which executes the functions that adjust the total agent population.

7.1.2.2 State Variables

Agent State Variables Agents represent human park visitors, and each individual agent is described through a set of state variables. These are (Table 7.1):

- Group Size: The number of individual humans represented by this particular agent. Previous studies (Costa, 2010, Jazwinski and Walcheski, 2011) have demonstrated that people in public spaces often appear in groups, with most sizes between 2 and 5 people per group, 2 the most often. Relationships between park visitors are not included in this model, members of the same group are assumed to exhibit uniform behaviour, and so are modelled as a

Variable	Value Type
Group Size	integer
Location	3d Vector
Interaction Distances	float
Movement Speed	float
Age	integer
Lifetime	integer
Current Activity	activity type
Activity Duration	integer

Table 7.1: Agent State Variables

single agent, with this variable informing on group size.

- Location: The position of the individual agent within the virtual space.
- Interaction Radii: The different distances at which this agent responds to. Other agents and elements within these distances will affect this particular agent's behaviour. Distance lengths are affected by group size, especially close interaction/personal space radius, which correlates with group size. Specifics for these radii are discussed in *2.2.3: Distances in Social Interaction* (Gehl, 1987, Ciolek, 1983).
- Speed: The speed at which this agent moves through space. Agents are assumed to use walking as their only means of transport. Movement speed is assumed to vary slightly per agent, around a mean of 1.5 m/s (Ishaque and Noland, 2008).
- Lifetime: The total time period this agent will exist in the simulation, representing the park user's visit length.
- Activity Type: Walk, Sit, Prepare-To-Sit, Prepare-For-Sports, Sports, Visit Feature, Exit.
- Current Activity: The activity the agent is currently engaged in. Can be one of the previously defined activity types.
- Next Intended Activity: The activity the agent intends to engage in next. Can

be one of the previously defined activity types.

- Activity Duration: The duration of the current activity/next intended activity.

Environment State Variables The Environment holds a number of state variables as well, which affect agent behaviour. These are:

- Terrain Type: Different types of terrain have an effect on agent behaviour. Paths are the preferred terrain to walk on, green areas are preferred sitting locations, water presents a limit for activities and movement, but not vision.
- Features: Main park features, eg trees.
- Attractions: Main attraction elements in the environment, might include fountains, restaurants, etc. Elements fixed in space that are known to have an attractiveness in terms of human activity.
- Time of Day: The model captures park activity throughout the day, and as such the time of day is considered as an independent state variable, ie. it is continuously incremented, cannot be affected by other variables.
- Day of Week: One of the seven different days of the week. Day types are grouped into weekdays and weekends, as these have been observed to attract different behaviour.
- Weather: This model deals with outdoor environments, and as such it expects weather conditions to have a significant effect on park use.

7.1.2.3 Scales

The different scales in the model (temporal and spatial) are approached thus: Time is considered in a continuous fashion, with one timestep in the simulation representing one second of simulated time. Agents act in an asynchronous fashion, on the basis that all of their potential activities will always last more that one timestep. Each agent is locked into execution of its current activity until it ends, and when the

activity duration comes to an end, the agent continues with the next action. Technically, all agents update at the same time in asynchronous fashion, however each agent is locked into execution of its current activity until it ends, and given the high temporal resolution, agents are assumed to not be updating any major states during most timesteps.

Space is considered as a continuous element in 3 dimensions (x, y, and z), with potential overlap of elements in the vertical dimension, eg. bridges. Extents depend on application/target space, but are generally found to be approximately 1km x 1km, with one distance unit in model space representing one metre.

7.1.3 Process Overview and Scheduling

The controller acts as the master element in all respects, controlling the high level functions of the model. The model described here is a perpetual model, by which meaning that the simulation is designed to run indefinitely, simulating/recreating real-world conditions and activities in a public space. The controller in this framework controls the modifiable environmental parameters accordingly (eg advances time, sets weather conditions) and more importantly controls the total number of agents in the simulation. These elements are adjusted/updated at a more infrequent rate (every 900 timesteps or seconds, thus every 15 minutes in simulated time).

The controller's aim is to have the correct number of agents in the simulation, as retrieved and evaluated against input from external sources. It adjusts the overall agent population accordingly, by either introducing additional agents at the beginning of the controller update, or by flagging older agents to execute exiting behaviour and remove themselves from the simulation (Figure 7.4). Furthermore, the controller records, collates, and visualizes core model behaviour, such as total population, aggregate activity, crowding, etc.

The environment is mainly a passive static element, meaning it has no control over its variables, the few which are modifiable are set by the controller. These include

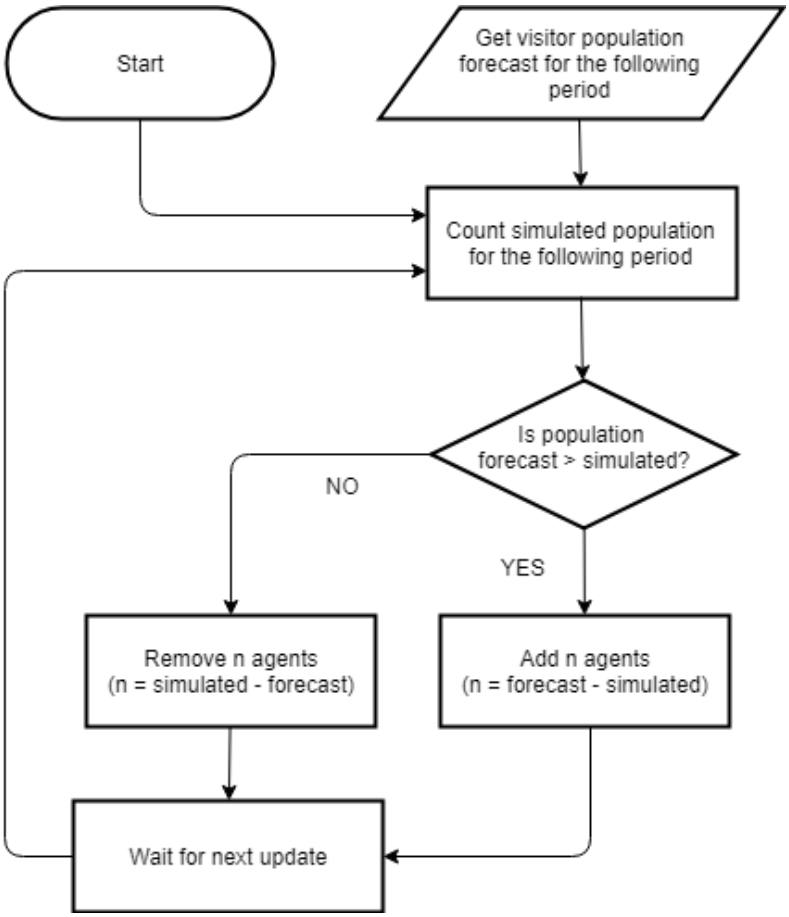


Figure 7.4: Model Controller Loop

accessibility to features and points of interest depending on time of day (eg opening hours of establishments in the area), or score penalties for specific activities/areas based on weather (eg. lawns become even more unfavourable walking terrain during unfavourable weather conditions).

The agent process overview is as such:

1. Agent is introduced into the area.
2. while the agent's age is less than their predetermined lifetime, the agent executes the following steps:
3. Agent decides on its next activity
4. If said activity requires preparation, the agent begins preparing (often an iter-

ative cost minimization process) until a condition is met.

5. When the agent has completed any/all required preprocessing tasks, it moves to the desired location and engages in the intended activity (plants themselves in space) for a predetermined duration.

6. At the end of the activity duration, the agent again decides on its next activity.

7. Once their lifetime is reached, the agent removes themselves from the area.

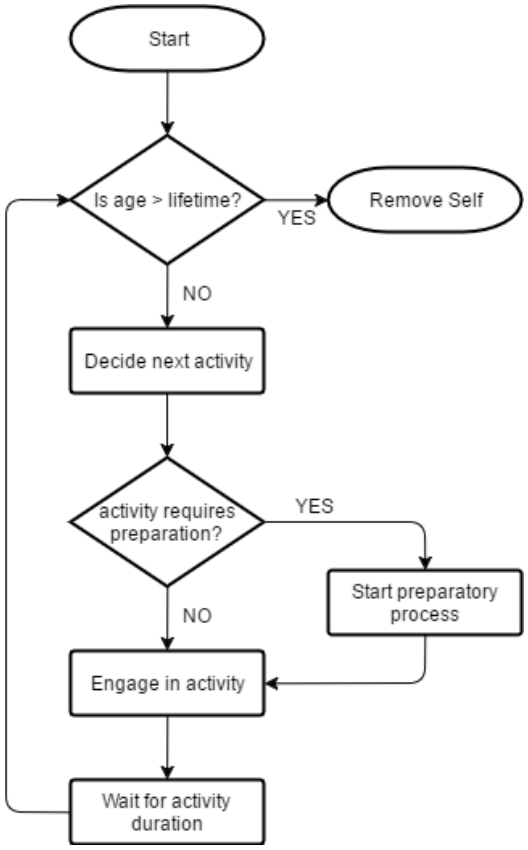


Figure 7.5: Agent Behaviour Flowchart

The agent activity decision-making process and is modelled as a Probabilistic Finite-State Machine (PFSM). At each activity decision step, the next intended activity depends on which activity the agent has currently completed. For example, walking activities are often followed by another walking activity. Stationary activities (eg sitting) are never followed by a stationary activity, but lead to a walking activity, either as a random walk, or by following a path to the exit.

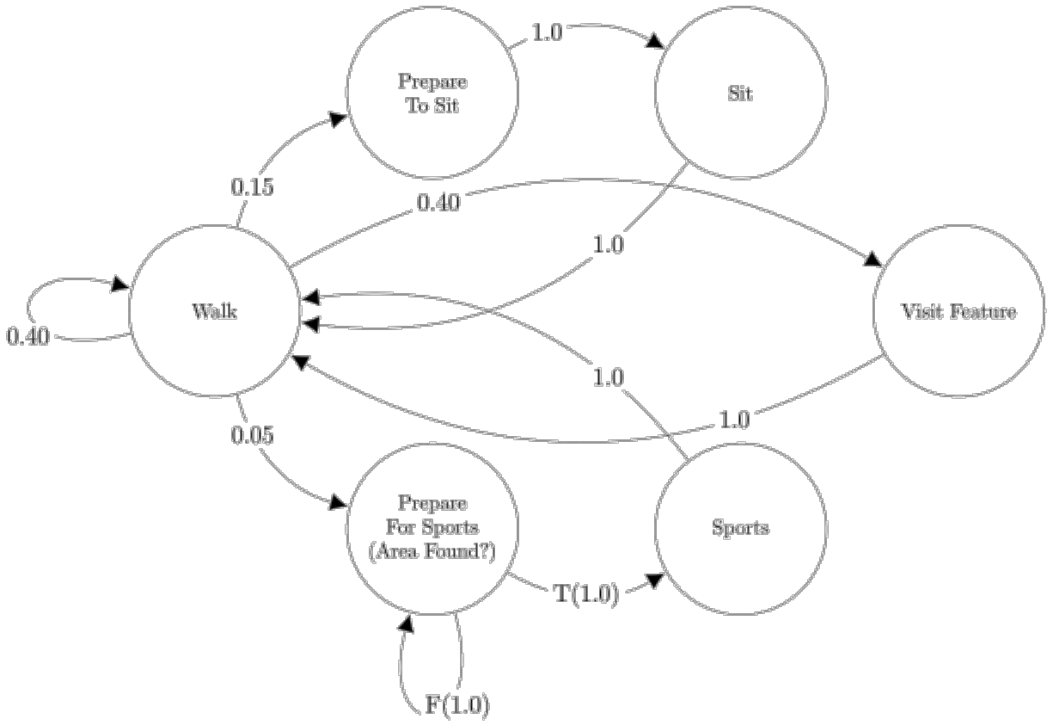


Figure 7.6: Agent Decision Process as a Probabilistic Finite State Machine

7.2 Design Concepts

Basic Principles Foraging Agent Behaviour: Each agent at its core exhibits a foraging behaviour, by which it pursues the maximization of some element/variable. This cost variable often represents comfort. Essentially each agent attempts to find the optimal location for their intended activity, with the definition of optimal depending on environment parameters, as well as other agents' established behaviour, ie it is a local optimal (local both spatially, in the agents vicinity, and temporally, as agents spend a limited amount of time for precalculation/foraging). This principle is implemented exclusively at the submodel level.

Evaluation of PSU findings: The model aims to evaluate existing findings and observations on human social activity in public spaces. Therefore, it evaluates whether the proposed/observed behaviours, when implemented in a model at the individual level, produce realistic/observed aggregate behaviour.

Emergence Spatial Distribution of Activity at any point in simulation time is expected to be a product of environmental variables, as well as (and mainly) interaction of agents, highly dependent on agent preferences and simulation conditions.

Adaptation The agents generally lack any immediate reactive behaviour, ie they do not respond directly to changes in their environment. The main behavioural element is their attempts to identify and locate themselves in appropriately crowded locations, so that they have enough agents around them, but not too many.

Objectives The main objective of each agent entity is to identify the location with a good crowding score, ie, plenty of agents in the general vicinity, but with enough free space in a small radius, free of other sitting agents. Locations are sampled at random within the agent's vision range, and their scores are calculated as sums of nearby agents: other walking and far agents are counted positively, while stationary agents at a close distance are counted negatively.

Sensing Agents are aware of some environmental variables, such as time of day, day of week, weather conditions. They also employ vision, and they are able to detect other agents and the terrain around them. Sensing other agents is used to calculate location scores, while terrain inspection is used to identify potential suitability of locations in terms of terrain, as well as to aid in calculating a path to target location. Finally, at specific cases (and this is a minority) agents have global terrain knowledge, specifically when they are required to navigate to a far destination (outside their vision range). These cases involve fixed locations, such as a feature/point of interest, or an exit.

Interaction There is minimal interaction between agents, as agents do not exchange or share any type of resource in the model. Agents are only aware of other agents insofar as crowding is concerned.

Stochasticity The majority of agent state variables and submodels are assumed to run on random functions, with variables drawn from constrained random and/or

normal distributions. These include speed, lifetime, intended activity, activity duration, etc, but also overall movement, which is modelled as constrained angular random walks on a weighed surface.

Observation The model controller records the majority of individual agent state variables, at multiple times in the simulation. Main variables of interest are the location of agents, their current activity, as well as aggregate or overall variables, such as total population.

7.3 Details

7.3.1 Initialization

The aim and purpose of this ABM, as described here, is to function in a perpetual fashion, simulating human activity in a fixed real-world location, as it is currently exhibited (i.e. in real-time). In this regard, it should not be concerned with initialization, as ideally, it is initialized once, and consequently runs indefinitely. Additionally, state variables and other parameters are constantly redefined during run-time, as informed by external sources. The majority of parameters are either set by external sources (either datasets, or models), or are static in nature, and therefore always the same. Therefore, it can be considered that the initial state of the model is known, with the state variables and conditions being those observed in the real-world location at the current time, and that the *Current agent population* is estimated by the forecast sub-model (discussed in section 5.2), based on time of day and weather conditions.

One aspect of the model that is potentially not set by external sources is the specific agent state variables (e.g. location, lifetime, activity, activity duration, etc). In cases where visitors are expected to be in the area during model initialization (i.e. initializing the simulation within park open hours), then these attributes are drawn randomly from a normal distribution, as discussed later in this chapter (subsection 7.3.3). This is in addition to normal agent entity initialization, as entities can

be assumed to be engaged in an activity, and be further in their lifetime. Otherwise, initialization of agent state variables is similar to when a new agent is introduced into the normal flow of the simulation.

7.3.2 Input Data

The model requires input from multiple external data sources, feeding into it in real-time. These datasets and input feeds are discussed in detail here, according to the model aspect they apply to.

Regarding visitor population/agent entity population, external input is used both to drive the overall population in the near-future, and to evaluate recent outcomes of the model. For near-future estimation, the predictive linear regression model of social media-weather is used, as discussed in *Section 5.2.2: Total Visitor Volume Approach*. This predictive model offers an estimate of near-future activity in the area of interest, calculated as an outcome of weather conditions and temporal characteristics. It essentially is fed into this ABM as a single integer value representing the total expected number of visitors for the next 900 seconds (15 minutes). For evaluation of recent activity, geolocated social media events originating from the area of interest within the last period of interest are collected, and compared to the predicted amount during the previous prediction phase. This comparison is used to evaluate the model in terms of overall performance, and is further used to correct the model.

In terms of temporal characteristics, a reliable external reference is used to inform the model of the current time, date, day, etc at the area of interest. For this purpose, either System Time as retrieved from the computer running the simulation can be used, or an online tool/web-service can be queried, to retrieve the current time. An online resource is more favourable, as it provides an approach less prone to errors.

Weather conditions play an important part of the simulation. These are retrieved automatically at each controller update phase, and include both current conditions and

a prediction of near-future conditions. An online weather forecast service is used[1], which collates weather data from multiple sources and weather forecast models, and which is queried through its Application Programming Interface (API) for weather conditions at the location of interest.

7.3.3 Submodels

7.3.3.1 Population Control

The controller entity is responsible for high-level processes, including controlling the agent population in the simulation. The model discussed here aims to provide a perpetual simulation, in which simulated entities correspond to and reflect real-world conditions at the current time. The mechanisms for controlling the simulated agent population are discussed in this section.

As discussed earlier, the controller updates at fixed intervals, every 15 minutes in simulated time. During these updates, the controller attempts to correct any differences between actual public space user population and simulated agent population. As a first step, it receives an integer value reflecting the number of visitors/users expected to be in the area of interest during the coming period (until the next update) on average. This is a predicted value, calculated from an external model (see *Section 5.2.2: Total Visitor Volume Approach*). Additionally, the forecast population value for the previous period is compared with the actual recorded population for the previous period, in a validation step, and the difference is added to the predicted value for the next period. This final non-negative integer value is then considered to be the target maximum agent population (P_A) for the following period.

Next, the controller calculates the projected unmodified current population for the coming period (P_S). This is calculated as a sum of all agents currently in the simulation, subtracting the number of agents planning on exiting before the next update, and adding any agents already flagged for exiting during a previous update that have

[1] forecast.io

not yet exited (so as not to double-count agents flagged for exiting).

A comparison is then made between P_S and P_A, so that $P_{Diff} = P_S - P_A$. If P_S is found to be less than P_A ($P_{Diff} < 0$), the controller starts introducing new agents into the simulation, equal to the difference between the two. Inversely, if P_S is found to be larger than P_A, the controller flags agents for exiting[2], equal in number to the difference between the two. The controller iterates through the list of current agents, in chronological order, so that older agents are flagged first, until the required number of agents has been flagged. The code implementation of this process is presented in detail in section B.2.

7.3.3.2 Agent Movement

Agents in the model have movement capabilities, enabling them to navigate within space. Two different navigation algorithms have been incorporated in the model, each serving a specific function: a Randow Walk Algorithm (RW) variant, and a Shortest Path Algorithm (SPA). In order to run, both of these algorithms require an abstract structured representation of space as a graph. Given that space is treated as continuous in 3 dimensions in this model, a navMesh has been implemented to represent navigable space in graph form (Figures 7.7, 7.8).

The navMesh forms the basis for all path-finding and navigation tasks performed by the agents in this model. Agents calculate shortest paths to their targets using the A* path-finding algorithm (Hart et al., 1968) over the navMesh, using Unity's implementation (Unity Technologies, 2017). The assumption that any required path should be the shortest one is derived from observations in relevant literature (Gehl, 1987, Whyte, 1988, Gärling and Gärling, 1988, Jazwinski and Walcheski, 2011, Bitgood and Dukes, 2006), where it has been noted that in public open spaces, once a pedestrian has a target location, they will prefer the shortest route.

The navMesh is split into different areas, each area associated with different move-

[2]Flagging an agent for exit essentially means that in its next activity decision, the agent will start its exiting process

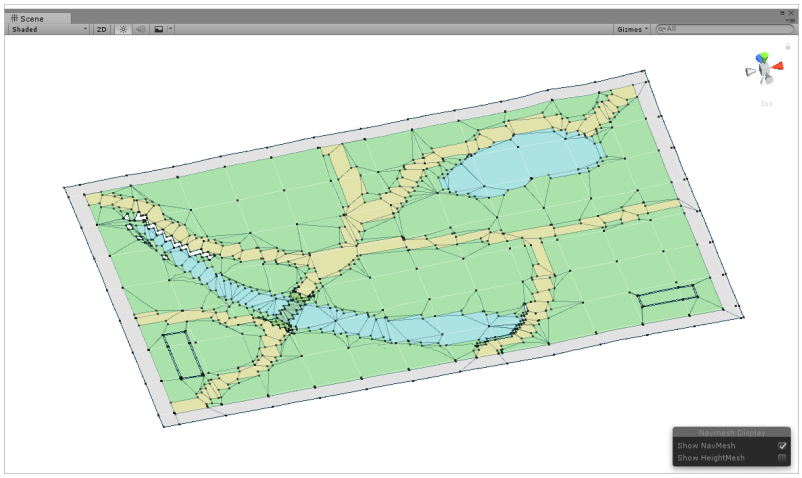


Figure 7.7: Area NavMesh

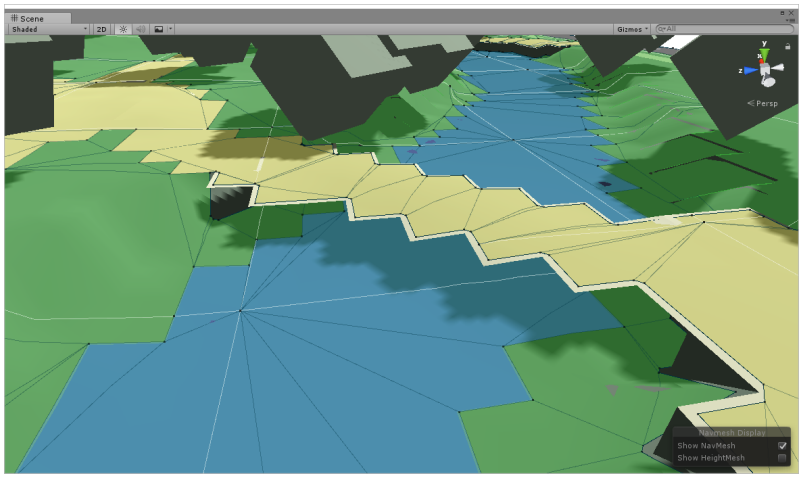


Figure 7.8: Area NavMesh Closeup: Area Overlap

Area	Cost
Path	1
Green	2
Water	-
Road	5

Table 7.2: NavMesh Area Costs

ment costs. The 4 main area types are (Table 7.2): Paths, are the default walking areas. These areas have the smallest traversing cost (1), and are thus preferred by agents. Green areas, lawns, etc. These areas are traversable, at an increased cost (2). Water includes bodies of water. These areas are non-traversable. Road includes roads allocated to vehicle traffic, they constitute the least preferred movement areas.

The Randow Walk Algorithm (RW) variant used in this model is an angular-constrained random walk. It is used as a heuristic for wandering behaviour in parks. Given the nature of the spaces considered in this thesis, focussing on parks, the majority of activities taking place in these areas may be considered leisure activities, or at the very least not target-based activities (i.e. they do not aim at optimizing a path to a specific target, such as commuting tasks might be). With this in mind, it is reasoned that any agents not actively moving towards a specific fixed target will assume a wandering behaviour, navigating seemingly randomly throughout the area.

In programmatic sense, the implementation uses a vision cone for the agent, similar to the RW implementation by Penn and Turner (2001), with the main difference being the absence of a precalculated visibility graph, instead using synthetic agent perception for identifying potential destinations at the time. An agent will pick a new valid location at random within its view distance (Social Distance or more), and calculate the shortest path using the navMesh to that point. This particular implementation of wandering behaviour includes a directional angle constraint, so that the new location must satisfy the parameter that the angle on the horizontal plane between the agent's current forward direction vector and the vector from the agent to the new location is smaller than the agent's field of view[3] (Figures 7.9, 7.10).

In addition to the RW variant, in some instances agent movement is required to cover large distances. Such instances involve specific features in the area, with a fixed location, that might be the target of an agent's action. These include navigating to an attraction, facility or amenity, or moving to a gate to exit the area. This form of long-range path-finding again implements the A* SPA, calculating the shortest path to target location, with the only difference that the target location can be anywhere in the area (outside the agent's view range, angle of view, etc) (Figure 7.11).

[3]Essentially, making sure the new direction is broadly 'in front' of the agent, thus mostly eliminating backtracking and orbiting the same location

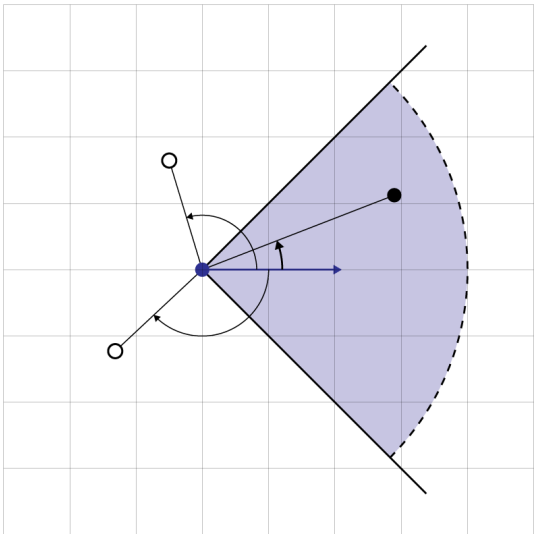


Figure 7.9: Angular-Constrained Random Walk: The blue arc represents the agent's field of view. White circles highlight random target locations outside the agent's field of view, the black circle highlights a valid location.

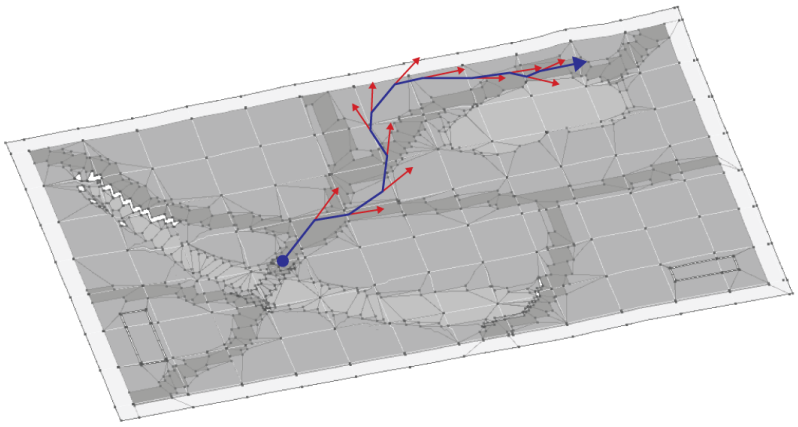


Figure 7.10: Angular-Constrained Random Walk - Resultant Path

Agent movement speed is derived from literature, is set to be 1.5 m/s on average (Willis et al., 2004, Ishaque and Noland, 2008), and is considered a constant agent parameter, meaning it stays at the same value throughout the agent's lifetime. Individual agent speeds are drawn from a normal distribution with a mean of 1.5 and standard deviation of 0.15, with final values constrained between 1 and 2. Although literature suggests that speed is inversely correlated to group size (Gärling and Gärling, 1988, Willis et al., 2004), and furthermore group size is an agent attribute included in this model (as discussed in the following section, subsubsection 7.3.3.3), this correlation between group size and group speed has not been

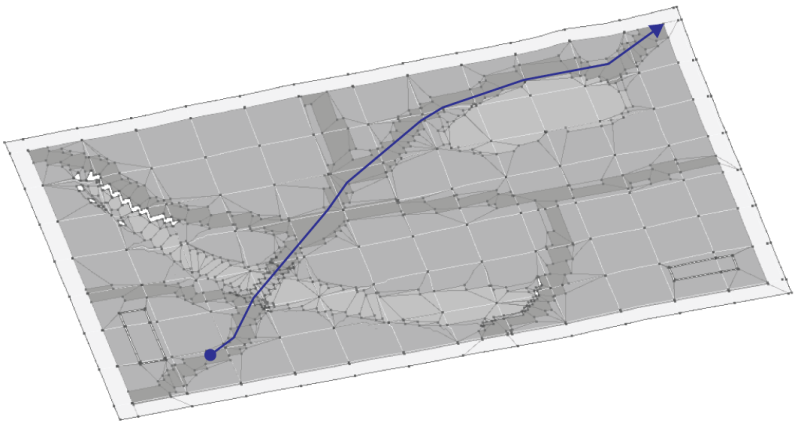


Figure 7.11: Long Range Path-Finding

implemented in this model.

7.3.3.3 Agent Group Size

Relevant literature notes that people in public spaces are most often encountered in groups (Costa, 2010, Jazwinski and Walcheski, 2011). Following from these observations, individual agent entities in this model do not correspond one-to-one with actual park visitors, but rather represent groups of people, as suggested by their 'Group Size' state variable. Groups in the model are assumed to be inseparable, and are therefore represented as a single agent, and furthermore 'Group Size' is considered to be constant throughout each agent's lifetime (cannot change during the course of the simulation, but can and does vary between different agents). The 'Group Size' parameter essentially affects the way an agent is perceived by other agents when it is being seen and counted by any function: each agent will be counted a number of times equal to its 'Group Size' parameter, e.g. when calculating relative densities, an agent with a 'Group Size' of 3 will count as three individuals.

Observations from literature state than pairs are the most often encountered group size. This observation seems to be further verified by a visitor survey at one at the areas of interest of this work (Ipsos Mori, 2015a), which identified parties of two as the majority of cases. Taking into consideration these observations, group size for

the agents is calculated as such: valid group sizes are considered between 1 and 4 people (groups of size 5 and over were a rare occurrence (<0.01), and will not be considered in the model). Agent group size is calculated at random during agent initialization, with probabilities as shown in Table 7.3.

Group Size	Probability
1	0.42
2	0.488
3	0.046
4	0.046

Table 7.3: Agent Group Size

These probabilities have been derived at based on visitor surveys at Hyde Park, London (Ipsos Mori, 2015a), and seem to agree with previous observations. Following from these probabilities, every 1 agent corresponds to 1.72 visitors. This grouping reduces computational load, as it reduces the number of individual entities needed to exist in the model.

7.3.3.4 Agent Interaction Distances

Human socio-spatial behaviour has been identified in literature to vary greatly, depending on the distance between individuals during interaction, or inversely, that specific distances are obeyed depending on the type of interaction taking place between two or more people. Multiple different interaction zones have been identified (Hall, 1966a, Ciolek, 1983, Gehl, 1987), and have been discussed extensively in *Section 2.2.3: Distances in Social Interaction*. A summary of these findings is offered here. Research seems to agree at an upper distance threshold that encompasses inter-human interaction, observed to be approximately at 100 meters. The degree of familiarity and intimacy of interaction appears to be inversely correlated to interaction distance, with interaction between close friends and acquaintances taking place within 7.5 to 10 meters. From this it also follows that non-friends within this distance are generally avoided, i.e. between strangers, such distances are generally observed to be maintained. Interaction between strangers or in for-

mal circumstances takes place in distances between 10 and 70 meters, with these distances including the act of spectating as an interaction. Between distances of 70 and 100 meters, others are acknowledged as people within the same general area.

In the context of this model, these observations have been adopted and simplified in order to inform the agent behavioural framework concerning rules of engagement with other agents. The upper limit has been adopted exactly, at 100 meters. At the other end of the spectrum, the default personal distance threshold has been defined at 10 meters. The distinction between social interaction and spectating distance has been dropped, and instead all interactions between 10 and 100 meters are considered as similar, as the final agent interaction and behaviour framework does not require this level of fidelity. Personal distance may be considered to be higher, dependent on agent group size, up to 15 meters. Therefore, overall agent interaction distances are defined as follows:

- Personal Distance: 0-10 m
- Social Distance: 10-100 m

7.3.3.5 Agent Lifetime

The vast majority of people in public spaces only spend a specific, predetermined amount of time in any one space, as public space use is considered as ephemeral space. This is generally true for visits to parks as well, although with the difference that visits might have a longer duration, as parks are considered as spaces for leisure activities. Given the above, agents in this model have a predetermined lifetime in the simulation, calculated at random during agent initialization. The values along with probabilities for different durations are taken from park visitor surveys, carried out for one of the areas of interest in this work (Ipsos Mori, 2015a). Overall visit durations are allocated at random with probabilities as shown in Table 7.4.

Under these durations and probabilities, an average park visit is considered to last 80.7 minutes.

Agent Lifetime (minutes)	Probability
0-30	0.16
30-60	0.24
60-120	0.39
120-180	0.16
180-240	0.04

Table 7.4: Agent Lifetime

7.3.3.6 Agent Activity Duration

The formula used to calculate how long each stationary activity should last takes into count the following parameters: the duration (in frames/update ticks) of an average walk action t, the overall probability P_S that at any point in time an agent will be involved in any stationary activity (calculated as the sum of all activity probabilities), and the time D_P spent preparing for the next stationary activity. The consideration behind this calculation was that on average in the model, if the agents have a probability $0 < x < 1$ of engaging in a particular activity, then they will also spend x percent of their total lifetime engaged in this particular activity (on average across all agents). The final value for the duration of the stationary activity D_S is expressed as a factor of t, so as to make it applicable over different models:

$$D_S = t * mod * c$$

Where c is a constant and mod is the modifier to be applied to t. Through trial-and-error, mod was set to

$$mod = \frac{v_1 + v_2 - 1}{v_1 - 1}$$

where $v_1 = 1/P_S$ and $v_2 = D_P/t$, and $c = 1.5$.

7.3.3.7 Agent Activities

The different activities an agent might engage in during their lifetime in the simulation have been presented earlier. They are listed in Table 7.5.

Movement activities implement the Randow Walk Algorithm (RW), as has been

Activity	Activity Type
Walk	Movement
Prepare to Sit	Precalculation
Sit	Stationary
Feature Visit	Stationary
Prepare for Sports	Precalculation
Sports	Stationary
Exit	Movement

Table 7.5: Agent Activities

described earlier. Precalculation activities implement the 'Walk' behaviour at their core and run additional scanning algorithms during their execution; essentially the base 'Walk' behaviour is used in order to allow the scanning algorithms to cover a larger area/increase the sample size. Stationary activities require a target location to be specified, and are carried out in two phases: the first phase involves moving to the target location, and the second phase involves the agent engaging in the activity with a fixed position in that location, for the duration. The 'Exit' activity functions the same way as a 'Feature Visit' activity, with the exception that when the agent reaches its destination, it is removed from the simulation, instead of engaging in an activity.

Activity duration is calculated as a function of the average duration of activities the agent has participated in so far. Some activities (such as walking) have a duration as a resultant variable (duration / speed), while others (such as how long a stationary activity will last) rely on the duration being known beforehand. Agents always start at a 'Walk' state, ensuring that the average duration variable is defined. The following sections will discuss individual agent activities, and how they are implemented in the model.

7.3.3.7.1 Walk Agents move around the area using the wandering behaviour discussed in 7.3.3.2: Agent Movement. This behaviour is considered the default state of agents, with the highest probability. Once the 'Walk' has been triggered, the process is as follows: The agent picks a random location on the ground, within its field of view. This location it then examined, to verify whether it is on navigable terrain.

If not, a new location is chosen at random. If the location is on valid terrain, a path from current location to target location is requested. If no valid paths exist, the process is reset and a new location is chosen at random. If a valid path exists, the length of the complete path is compared against the straight-line distance between origin and destination. If path length is found to be longer that 4 times the straight-line distance, a new location is chosen at random and the process resets. This distance check is performed to ensure agents are not attempting to navigate around a large obstacle (e.g. a long narrow river with a single bridge). If the path is valid, the agent sets it as its current path travels along until it reaches the end of the path, at which point a new activity is decided.

7.3.3.7.2 Prepare-To-Sit Prepare-To-Sit is a pre-routine, is almost always followed by the 'Sit' activity, and it deploys a scanning behaviour for the optimal location to sit. Once triggered, it will request a random walking path, as per the 'Walk' routine. When it completes, the duration for the following sitting activity is also calculated, using the formula presented in subsubsection 7.3.3.6. The scanning process ends after a predetermined length of time, expressed as a multiple of the average walking duration. During this scanning phase, 'Walk' behaviours are being triggered until the required time has passed.

The scanning process itself involves a form of agent vision, which is implemented using collision detection algorithms through a physics engine. The process is illustrated in Figure 7.12. At fixed short intervals (potentially at every update), a location on the ground is chosen, within the agent's current field of view. The target location is verified to be on navigable terrain, otherwise a new location is chosen. If the location is navigable, a virtual sphere is placed, centered at the target location, with a radius of the agent's social distance (100 m). All physical geometries of type 'agent' that overlap the sphere are returned in a list as potential entities of interest. The length of the list (number of other agents visible from this location) is considered to be this location's score. However, if other agents currently engaged in a sitting activity are found to be within the personal distance (10 m) from target

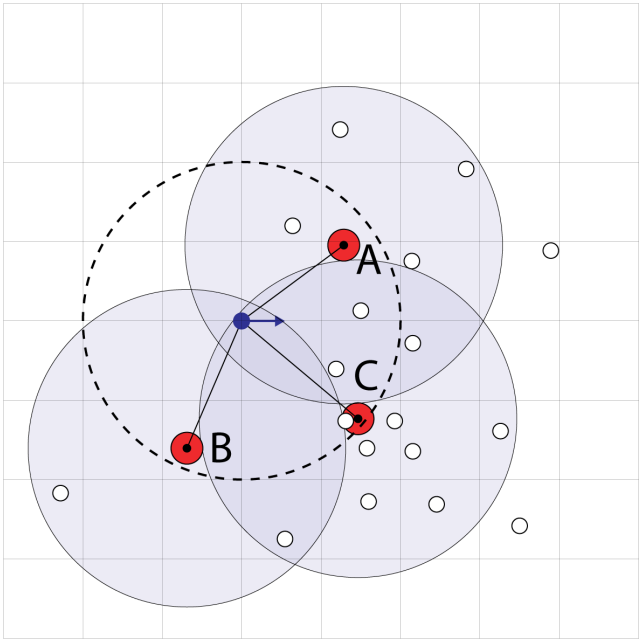


Figure 7.12: Agent Prepare-To-Sit Scanning Process. For 3 potential locations A, B, C, scores are calculated as: A=7, B=2, C=11. C however is discarded, as other agents would fall within the scanning agent's personal radius. Therefore position A is chosen as the winning location.

location, the location is discarded. At the end of the scanning process, the location with the highest score is considered as the optimal. A path to that location is calculated, and the agent sets it as its current path. Once the target location is reached, the agent checks whether it has been flagged for exit, or whether its lifetime has been completed, in which cases it skips the sitting activity, and starts its 'Exit' behaviour. Otherwise, it engages in a sitting activity.

7.3.3.7.3 Sit The agent plants itself at its current location, and changes its state to 'Sitting'. The duration for this sitting activity has been calculated already, as two-thirds of the overall preparation and actual sitting activity. The agent stays at this location for the duration, at the end of which an exit check is performed, otherwise a 'Walk' activity is triggered.

7.3.3.7.4 Feature-Visit The Feature-Visit sub-process involves the agent visiting a predetermined fixed location in the area of interest (e.g. a restaurant, an attraction, etc.). The agent picks one location from a pre-compiled list of points of in-

terest in the area (if no such features exist in the simulation, this behaviour is not implemented). Next, a path is calculated to that location, and is set as the current agent path. Once the agent reaches its destination, it engages in a sitting activity, by planting itself at the location, and changing its state to 'Sitting'. In a similar fashion to the sitting activity, activity duration is calculated using the formula in subsubsection 7.3.3.6, and the time taken to move to the feature is used a the time spent preparing for the activity. At the end of the sitting activity, an exit check is performed, otherwise a 'Walk' activity is triggered.

7.3.3.7.5 Prepare-For-Sports Prepare-For-Sports is a pre-routine, and is almost always followed by a 'Sports' activity. It employs a scanning behaviour, similar to the 'Prepare-To-Sit' activity, allowing agents to identify potential locations appropriate for sports activities. The 'Prepare-For-Sports' behaviour overlays a 'Walk' and a scanning behaviour, utilizing agent movement to allow the scanning algorithms to cover a larger area. For the duration of the scanning phase and until a suitable location has been identified, walk destinations and paths are generated continuously, allowing the agent to wander throughout the area. Sports activity duration is calculated using the formula in subsubsection 7.3.3.6. In contrast to sitting and feature-visit activities however, the sports preparation activity only ends once a suitable location has been found (or if none are found, when the agent exceeds its lifetime). Therefore, this preparatory routine can last for a significant duration, and subsequently the sports process that follows it will have a long duration as well.

The scanning process involves a form of agent vision, implemented using collision detection. The process is illustrated in Figure 7.13. At fixed intervals, a random location on the ground is chosen, within the agent's current field of view. The target location is checked to be on navigable terrain and accessible from the agent's current location. If the location is found to be valid, a feature check is performed on the surrounding area, up to a fixed radius (reflecting the playing field) from the target location. This check returns all geometry of type 'feature' (e.g. trees, buildings, furniture), terrain types 'water' and 'path', as well as agents of state 'Sitting'. If

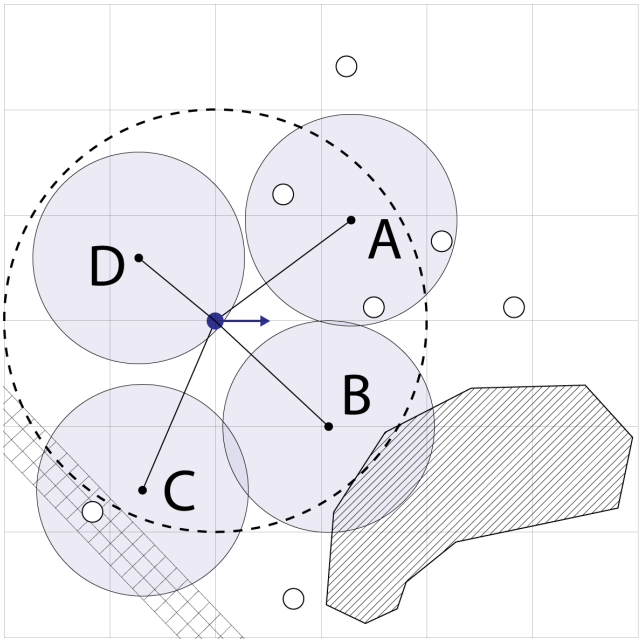


Figure 7.13: Agent Prepare-For-Sports Scanning Process. For 4 potential locations A, B, C, D: A is discarded as too crowded (3 other agents detected). B is discarded due to overlap with features. C is discarded due to overlap with path geometry (a single agent within the area is acceptable). D is a valid area.

none of the above are identified, the location is considered valid. Up to a single agent of state 'Sitting' can be detected and the area will still be considered valid. Agents in movement states are considered valid, as they do not occupy the potential playing field. Furthermore, agents in a 'Sports' state are also considered valid. However, any type of static geometry (features and/or terrain types) will render the location as invalid. The scanning process is repeated until a valid location is found. Once such a valid location is found, a path is calculated, the agent sets it as its destination, and moves towards it. Once the location is reached, the agent performs an exit check, if it returns false, the agent engages in a sports activity.

7.3.3.7.6 Sports The agent plants itself at its current location, and changes its state to 'Sports'. The duration sports activity has been calculated already, as the two-thirds of the overall sports activity (including both the actual activity and the preparation phase). The agent stays at this location for the duration, at the end of which an exit check is performed, otherwise a 'Walk' activity is triggered.

7.3.3.7.7 Exit An 'Exit' activity is triggered at the end of the agent's lifetime, or when it has been flagged by the controller for exit (to reduce overpopulation). Similar to the 'Feature-Visit' activity, the agent picks one target exit location (a gate) from a pre-compiled list of gates (if no such list exists, the agent is immediately removed from the simulation). Next, a path is calculated to the destination, and it is set as the current path. The agent then follows the path, and at the end of which is removed from the simulation.

7.4 Summary

This chapter presented the ABM of PSU developed in this work, using the revised version of the Overview, Design concepts, and Details (ODD) paradigm (Grimm et al., 2010). The overall aim of the model is to estimate user activity in parks at high temporal fidelity both spatially and temporally. In other words, its main outcome is a continuous simulation of park visitors capturing individual activities and their locations in the area of interest at a temporal resolution of one second.

The model consists of three core entity types: A static environment constructed using 3D mesh geometry, a single task scheduler (the controller) which performs simulation-wide tasks, such as time-keeping, agent population control, calculating run-time model performance statistics, and input-output functions, and finally the agents, synthetic autonomous entities representing park visitors that interact within the virtual environment based on predefined stochastic rules and conditions of their local environment.

Agent behaviours and decision trees were further presented in more detail: agents are introduced into the simulation, perform a continuous behaviour loop using a stochastic process implementing a Probabilistic Finite-State Machine (PFSM), and exit the simulation once their allocated lifetime has passed. The behaviour loop consists of five core behaviours (Walk, Sit, Feature Visit, Sports, and Exit) and two precalculation activities for two of the core activities (Prepare-for-Sit and Prepare-

for-Sports). Agent initialisation variables (including speed, group size, lifetime, and interaction distance) are drawn at random from pre-set value bins using fixed probabilities, as defined through relevant literature. Agent movement is implemented using two distinct algorithms: an angular-constrained random walk to simulate wandering behaviour, and a shortest-path using the A* algorithm to handle navigation to fixed locations. Agent vision was implemented using collision detection through the use of a physics engine. Finally, agent interaction was implemented in a form of scavenging behaviour for specific activities (specifically Sit and Sports activities), in which individual agents planning on engaging in such activities would sample locations in their vicinity, looking for the optimal location for each activity as defined by the presence of other agents.

For clarity, discussion focussed on the technical aspects of model implementation and was presented using a sample area. The following part (Part III) will demonstrate model applicability by presenting the application of the model to two case studies of real-world locations, focussing on model calibration and evaluation, and will furthermore present methods for coupling the ABM of PSU with Real-Time Data (RTD), in order to develop *Agent-Based Models of Public Space Activity in Real-Time*.

Part III

Applications

Chapter 8

Case Study 1 - Hyde Park

This Chapter will discuss the first case study carried out in this thesis, Case Study 1: Hyde Park (CS1:HyP), which focusses on Hyde Park (HyP) in London, United Kingdom. It will cover aims and objectives of the study, datasets used, both regarding their collection and analysis, the development and calibration of the two submodels discussed previously, specifically a forecasting model of aggregate visitor activity, and a Spatial Disaggregation Model (SDM) of individual visitor activity, along with an evaluation of the overall process.

The chapter begins with an introductory section (*Section 8.1*), highlighting the aims and objectives as set out at the beginning of the case study. Furthermore, it introduces the area of interest, identified as Hyde Park, discusses the reasons why this area was chosen, and highlights its advantages as a candidate area for the first case study.

The following section (*Section 8.2*) focusses on datasets used in this study, which include both remotely captured Real-Time Data (RTD) and ground truth data. It discusses the various methods used for capturing relevant data and presents some initial findings and limitations regarding datasets used. The section is divided into two subsections, one focussing on RTD routinely collected using collector programs, the other on data regarding individual visitor activity in the park, collected via site surveys conducted at various times and dates during site visits.

Following that, in *Section 8.3* a discussion on the development of the forecast sub-model is offered. This section discusses how RTD was used to calibrate the aggregate activity forecast model, allowing for continuous short-term predictions of overall activity in the park, using weather and social media data.

Next, *Section 8.4* presents an extended discussion on the development of the SDM used to simulate individual visitor activity in the park. It focusses on the application of the public space use Agent-Based Model (ABM) presented in *Chapter 7* on HyP, and covers the generation of the virtual environment, agent calibration, and output.

The second-to-last section (*Section 8.4*) offers a discussion on the evaluation of this first case study, both on the overall implementation and on individual components. Finally, the chapter concludes with a short summary, highlighting any particular limitations of the case study, and extracting any valuable findings.

8.1 Aims and Overview

The overall aim of this case study was to bring together all of the conceptual real-time simulation methodologies discussed in the previous chapters, and furthermore apply them to a real-world scenario, in order to test the validity of the overall model. This overarching aim was approached through a series of specific objectives, which helped to frame and guide the case study. The specific objectives this first case study set out to achieve were the following:

1. **Identification of relevant data sources.** Data source relevance was judged on how well a dataset captured Public Space Activity (PSA), its Real-Time (RT) characteristics, and its reliability and accessibility. The following data sources were ultimately used: Social Media (SocM) micro-blogging and photo-sharing platforms *Twitter* and *Instagram* were used as a proxy of visitor activity by capturing geotagged posts, weather forecast data from *forecast.io* (as an independent variable affecting visitor activity), and visitor activities with locations on specific days which formed the ground truth data.

2. **Development of appropriate data capturing methodologies.** Automated collection scripts were written in the Python programming language which collected SocM and weather data through web Application Programming Interfaces (APIs) every day. A similar methodology could be applied at a finer temporal resolution (e.g. 15 minutes or less), in order to have more recent, i.e. *Real-Time* information. Ground truth data was collected via surveys conducted during site visits.

3. **Development of a PSA forecast model, capable of performing in RT, and subsequent calibration of the model using available data sources.** A predictive model of total visitor activity was developed, as presented in a previous chapter (*Section 5.2*), and calibrated using SocM and weather data to continuously provide forecasts for visitor activity in HyP at 15 minute intervals.

4. **Development of a SDM using the ABM paradigm, to simulate individual visitor activity in the area of interest, capable of performing in RT. Subsequent calibration of SDM parameters.** An ABM was implemented to capture individual visitor activity in HyP, following the framework presented in *Chapter 7*. It was calibrated using data on individual visitors' activities, gathered through site visits.

5. **Evaluation of the overall RT model, as well as sub-models.** The forecast sub-model was validated against an independent subset of the collected data. Regarding the SDM, initial aims were to validate distribution of activities against an independent, real-time dataset, the geolocated SocM events. However, due to changes in SocM sources' handling of geolocation, this proved impossible, and ultimately the SDM was not validated against an independent dataset in this first case study.

A further, secondary aim of this first case study was to investigate the extent to which all other objectives could be achieved using solely publicly available datasets.

This restriction on data sources was established for two reasons: First, as an exercise and investigation into the extent to which public life is captured in datasets which are publicly available; in other words, whether physical public life is adequately mirrored in its traces in digital public life. Second, as a safeguard, in order to not restrict method application to exclusive datasets. This second reason was even more important given that this was the first case study undertaken and therefore method validity had not been established yet: this work needed to ensure that any methodologies developed in this work could (at least in theory) be applied to other areas as well as long as similar data sources were available for the other target areas, and it was decided that the best approach for this would be to only employ publicly available datasets.

Figure 8.1: Hyde Park Case Study Area Boundaries

About the area of interest: Hyde Park (HyP) is a metropolitan park, west of Central London, UK, maintained by the Royal Parks. It connects to the west with Kensington Gardens, also a park maintained by the Royal Parks, and together they form a large open area of London and of vital importance for green spaces. For this study, only HyP will be examined, for a number of reasons. First of all, it is of a roughly rectangular shape, with a side of over 1 km for a total area of approximately 127 ha,

with enough variation in landscape and features to host a large number of heterogeneous activities. Therefore its area is more than adequate in capturing potential variations in activity. Secondly, the park is surrounded by four carriageways, one on each side, with no vehicular traffic allowed within the park. These characteristics provide a well-defined set of properties for the purposes of this study, as the defined borders allow for a straightforward classification of visitors as being *in* the area, and furthermore the absence of motor traffic allows for a mostly uninterrupted study on human activity in public spaces, as visitors are free to move and use the entirety of the space. A final advantage of this location is its open-air characteristic, as HyP has very few structures or other tall features, which makes geolocation services easier to use (i.e. stable signal for GPS-enabled devices).

8.2 Data Sources and Analysis

Presentation of data used for this study, data sources, analysis and results.

8.2.1 Real-Time Datasets

As discussed in previous chapters (chapter 6), this work focusses on *Real-Time* simulation of activity in public spaces, and as such is concerned with datasets that capture park visitor/user activity *in Real-Time*, i.e. as it happens. Furthermore, as discussed in chapter 5, it aims to be continuously forecasting future activity in the short term, and therefore the majority of the datasets examined were considered due to their potential on constantly capturing and predicting activity in public spaces. This section will discuss all the datasets that were considered for capturing and/or forecasting park visitor activity in the first case study on Hyde Park.

The initial approach considered the use of data on transportation passenger journeys, with the aim to build a *Visitor Supply Forecast Model*, as discussed in subsection 5.2.1. The central idea was to make use of TfL's API on live train and tube arrivals, coupled with data on passenger exits at stations, develop a predictive model

that would continuously forecast the number of passengers exiting at each station around Hyde Park, and potentially extend to other modes of transport (e.g. buses), in order to have an estimate of new people arriving in the general area. However, as was stated in subsubsection 6.4.2.3, due to lack of real time datasets on passenger volumes and exits, the forecasts lacked any meaningful variation when disaggregated to quarter-hour resolution, as was demonstrated in Figure 6.24: 15-minute predictions for different days throughout the year resulted in only small offsets of the same daily curve. In addition to poor results, there was an issue regarding data availability: Although data retrieved via TfL's API can be considered publicly available, information regarding passengers and Oyster Cards is not publicly available. Therefore, if this approach had been enhanced with Oyster card data for more meaningful analysis, it would have invalidated the secondary aim of this case study, to attempt to simulate public space activity using only publicly available data. Due to the reasons discussed here, transport data was ultimately not used in this case study.

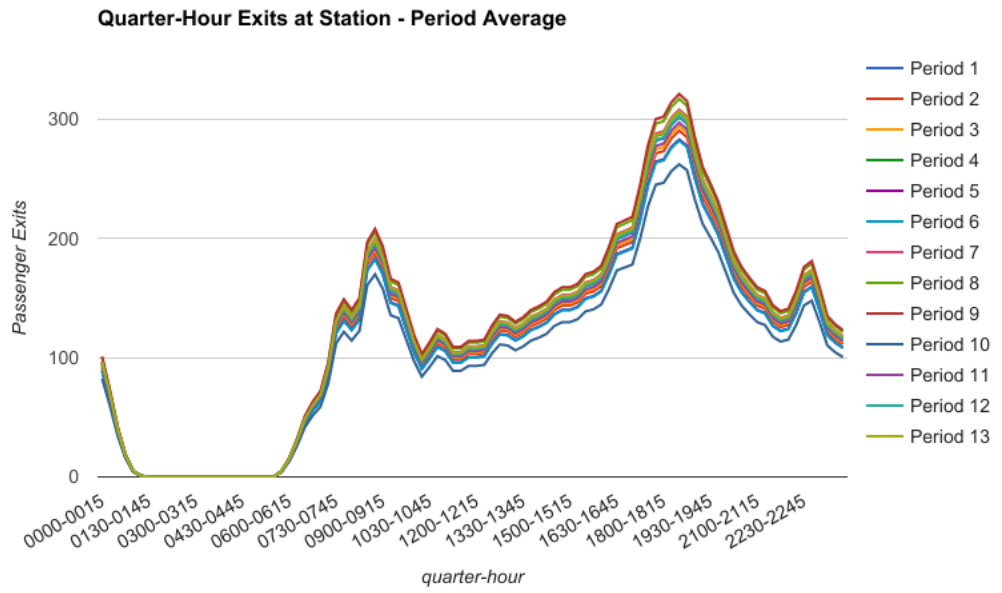


Figure 6.24: Passenger Exits at Station - Period Average (repeated from page 172)

8.2.1.1 Social Media Datasets

The second approach considered the use of Social Media (SocM) datasets, to be used as a proxy for real-time activity in public spaces. Specifically, 3 different platforms were considered: the social networking and micro-blogging service *Twitter*, the photo-sharing application *Instagram*, and the social networking service *Facebook*. All of the above platforms provide content to their users in a continuous, streaming fashion, and all three provide developers access via an API. Through their APIs, new content can be retrieved as it becomes available, i.e. *in Real-Time*, and further filtered via geolocation tags, to only include data from specific locations. Initial thoughts for the SocM approach considered the use of Instagram and Twitter feeds as proxies of actual activity on the ground, and Facebook to collect future planned events for anticipating and forecasting future activity volume. Of the three, Instagram and Twitter were used for the whole of the case study, Facebook was dropped from the list of potential datasets due to unreliability issues, as will be discussed in the following paragraphs.

SocM post collection for Twitter and Instagram was performed using automated scripts written in the Python programming language, as discussed in section 6.1. The scripts were set to run every day at 15 minutes past midnight, and queried social media services' APIs for geolocated posts originating in the area of interest, which were published any time during the previous day. A detailed presentation of actual code used is offered in Appendix A.3. The datasets were stored as daily records sorted chronologically using each individual post's timestamp in UNIX time. Although collection was performed once daily, the same approach and code can be used with shorter query windows, to collect e.g. posts published in the past hour, half-hour, or any other duration. As explained further in section 6.1, Instagram collection stopped on June 1st 2016, due to fundamental changes in the service's Terms and Conditions. For this reason, May 31st 2016 marked the end of the data collection period, as the remaining service (Twitter) could not compensate for the volume of data that was now missing. A time series plot of daily totals from the two

services for the duration of the case study (September 14th 2015 to May 31st 2016) is presented in Figure 8.3 (Daily zero values were due to collection failure).

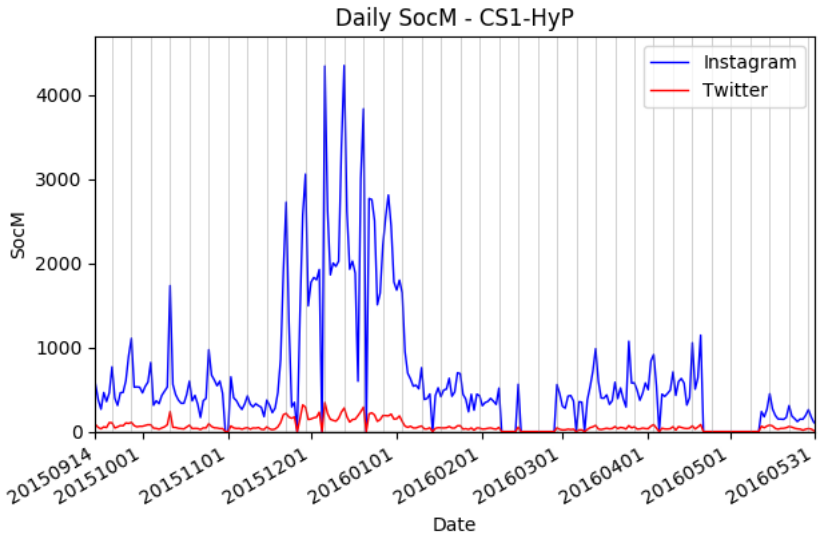


Figure 8.3: HyP SocM Daily Totals. Background vertical lines mark Sundays.

Planned events in the area were collected in a similar manner, using a script to access Facebook's API to retrieve planned events and number of attendees. The script used is presented in Appendix A.3. The script ran for a short duration as a pilot, from the beginning of data collection (September 14th 2015) until January 31st 2016. Data collected is presented in Figure 8.4, along with SocM daily sums from the two sources (Daily zero values in SocM were due to collection failure).

The Instagram feed consistently returns many more results, by a factor of 10 on average, compared to Twitter. This might be due to Twitter users not enabling location services as much as instagram users, and/or due to variance in platform usage in the park (Instagram's focus on visual information i.e. photos might be more favourable in a park than Twitter's text-based platform). Regardless of the resulting volumes, one thing to note is that both sources seem to exhibit the same peaks, and are generally consistent with one another. Finally, it is due to this difference in data volumes that the case study could not continue after Instagram data could not be retrieved any longer, as it constituted approximately 90% of all data.

The large block of increased SocM activity beginning on November 20th and end-

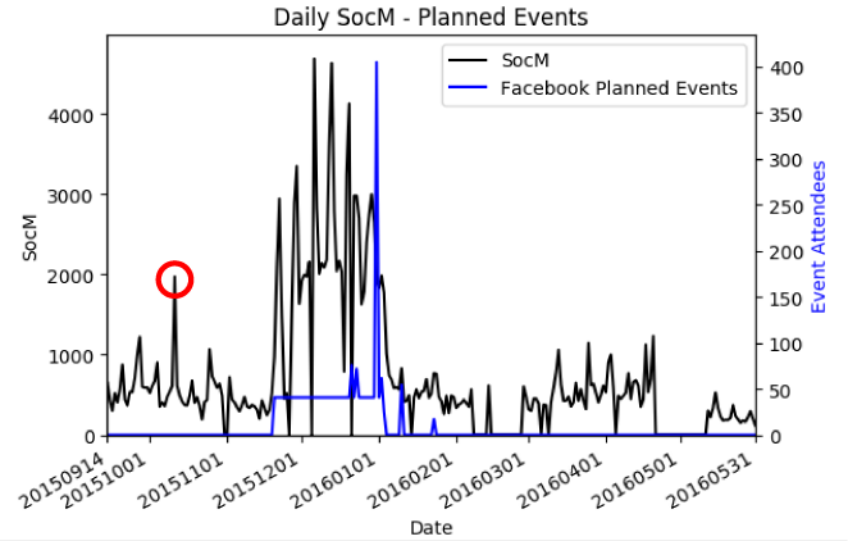


Figure 8.4: HyP SocM Daily Totals with Planned Events

ing on January 4th is due to an annual winter festival taking place on Hyde Park grounds, the *Winter Wonderland*. As expected, it consistently drew large numbers of visitors. The fact that the event was somewhat accurately reflected in Facebook planned events provided some initial support to the idea of using the Facebook API for using events as indicators of increased activity. However, the Facebook ecosystem imposed some further limitations regarding disaggregating the datasets. Although the event duration was captured adequately, attendees were only marked as attending the event once, without further information on date and time, which meant that the total number of attendees was averaged out across 6 weeks.

Further issues were identified with the Facebook API, specifically regarding meta-data and query results. This is highlighted in Figure 8.4 in red, on October 11th 2015. On that day, a half marathon run was organized starting at Hyde Park, which as expected drew a large crowd, however this was not reflected in Facebook's planned events. Further research identified the issue in the event being advertised on Facebook through a *page* type, rather than *event* type, which meant that it was not captured by the search terms. Although it is possible that this particular event might have been advertised previously as *event* type and subsequently removed, this instance highlighted a degree of 'messiness' in the Facebook ecosystem. This,

along with the fact that the API implemented frequent changes (the code used here stopped working soon after it was last run), made Facebook an unreliable source for real-time collection and forecasting.

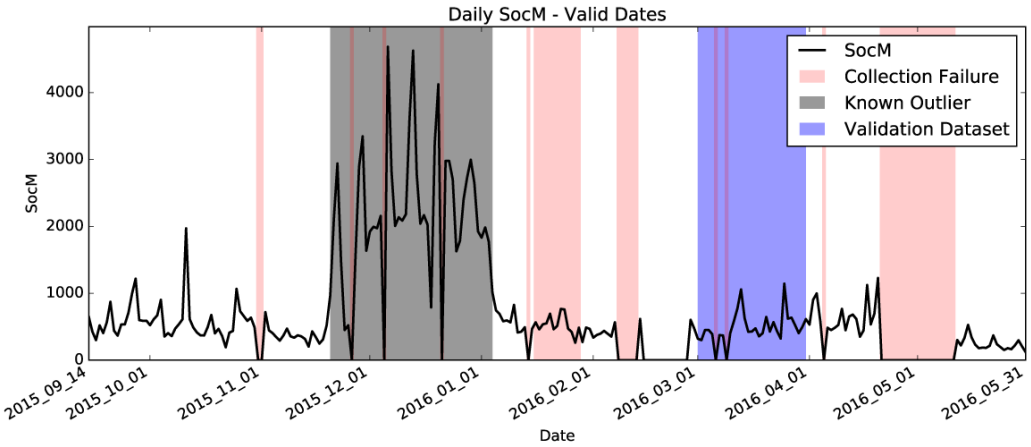


Figure 8.5: SocM Collection Valid Dates

A time series overview of SocM data collection for this first case study is presented in Figure 8.5. Overall, data collection covered a period of 261 days. Of these, automated collection scripts failed to run on 49 dates, and those dates were removed from the dataset. Furthermore, it was decided that increased activity during the 'Winter Wonderland' festival, spanning 46 days, would be treated as an extreme outlier, and was therefore discarded from the datasets. After removing these data points, the remaining dataset containing valid dates consisted of 169 days. Of these, 29 days (17.15% of total valid dates) containing all valid dates in the month of March 2016 were removed from the dataset and kept separately for validation purposes. The remaining 140 days, constituting 82.84% of total valid dates, were used to calibrate the *Forecast Model* discussed in section 8.3. A summary of the above statistics is offered in Table 8.1.

Having established the valid dates, the SocM dataset was temporally disaggregated to shorter durations, to be used in real-time forecasting. A duration of 15 minutes was chosen as the temporal resolution, for two reasons. First, at the onset of this case study it was assumed that transport passenger data would be used as well, which as discussed earlier was offered at 15-minute intervals, and so a similar time

	Days	Percentage
Total Dates	261	100.00
Known Outliers	46	17.62
Collection Failure	49	18.77
Valid Dates	169	100.00
Validation Data	29	17.16
Calibration Data	140	82.84

Table 8.1: SocM Data Collection Dates Summary

step was chosen for the SocM datasets, to allow for compatibility. Second, this value was considered as a good compromise between fidelity and meaningfulness, as it provided a duration long enough to carry substantial information on park activity, while at the same time it was short enough that it would capture variance throughout the day. 15-minute disaggregated values for a single day are presented in Figure 8.6 (black line).

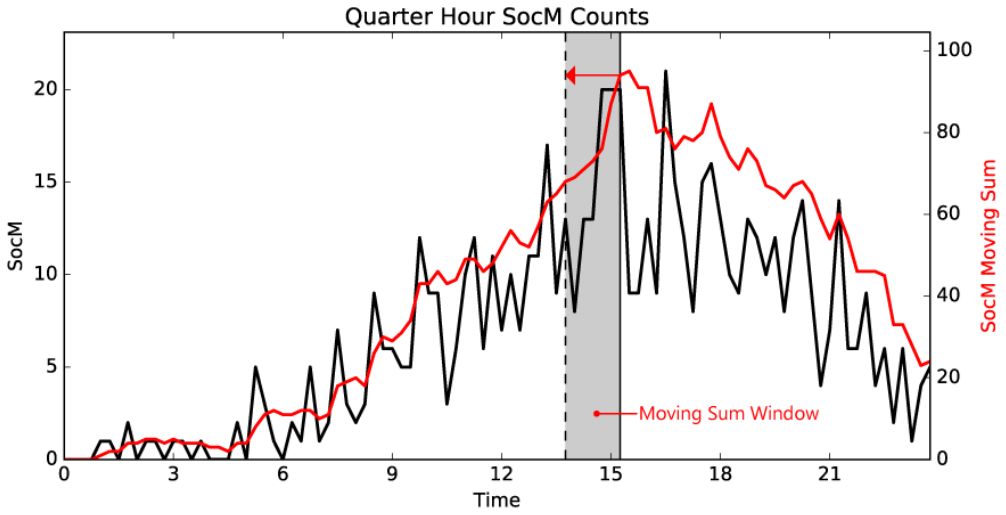


Figure 8.6: SocM Quarter Hour Counts for a sample day (27/10/2015). Quarter-hour values are calculated both in raw values as a simple count (black line), as well as smoothed values as a moving sum (red line)

It is evident from the graph that quarter-hour intervals capture the overall daily variation in adequate detail, with a steady rise in activity in the morning hours, peaking in the afternoon hours, followed by a steady decline into the evening and night. However, there is significant variance between time steps, as is evident by the sharp spikes in the graph. A smoothing function was applied to the full valid

dataset, in order to ease variation. The smoothing function employed a backwards 'moving sum' method, originating from a time step and moving backwards towards past time steps. The backwards moving sum was chosen in this case, rather than a rolling average often seen in time series analysis, due to the real-time nature of this work: At each point in time, the latest point for which data is available is at best the current point, i.e. no data about the near-future is available. Therefore, two options are available at this point, either averaging all points in timespan d centered at the point located at $d/2$, and therefore working with a time-delay of $d/2$ to real-world time, or summing all points in timespan d ending on the most recent point at 'now'. The second option was chosen in this case, as it was of importance to not compromise the real-time nature of this work. The moving window duration was set at 90 minutes, 6 times the dataset's time step, for two reasons: First, the average visit to Hyde Park has been observed to be approximately 80.7 minutes (Ipsos Mori, 2015a), and therefore a 90-minute window captures a typical visit, erring on the side of caution. Second, site surveys conducted at Hyde Park capturing visitor activity lasted just over 90 minutes on average (discussed in the next section) and were assumed to capture a still snapshot of park activity, and therefore maintaining a similar window on other data capturing approaches allows for the same assumption to apply as well: Essentially, a record of a SocM event signals the start of a new user's visit, which will last approximately 90 minutes, and therefore all visitors that arrived up to 90 minutes ago are considered to still be active in the area and contributing to current overall activity. As such, for each quarter-hour point, the final summed value of the past 6 quarter-hour counts was used as the current park visitor total.

8.2.1.2 Weather Data

Similar to Social Media data, information on weather conditions at Hyde Park was also collected at a high temporal resolution. This information was collected using the web API service 'darksky.net' (previously 'forecast.io), via an automated script written in the Python programming language. The code used is included in Ap-

pendix A.4, and was set to automatically make a request every day at midnight, capturing weather conditions for the previous day. As discussed in previous chapters (section 6.1), this API allows users to query weather conditions at a specific location on a specific date and time, returning either archived past data or forecast future data, and the response includes a wide range of environmental and weather parameters, at multiple temporal resolutions[1]. Data returned by the forecast.io API included daily and hourly resolutions. A subset of parameters considered relevant to park activity were extracted form the response and stored in daily JSON records. The extracted parameters were temperature, cloud coverage, wind speed, precipitation probability, and precipitation intensity, with measurement units presented in Table 6.2.

Parameter	**Abbreviation**	**Unit**
Hour	hr	#
Temperature	temp	C°
Minimum Daily Temperature	maxTemp	C°
Maximum Daily Temperature	minTemp	C°
Precipitation Probability	precP	percentage (0-1 range)
Precipitation Intensity	precInt	inch/hour
Cloud Coverage	cCov	percentage of sky occluded by clouds (0-1 range)
Wind Speed	wndSpd	mph

Table 8.2: Weather Parameters (repeated from page 153)

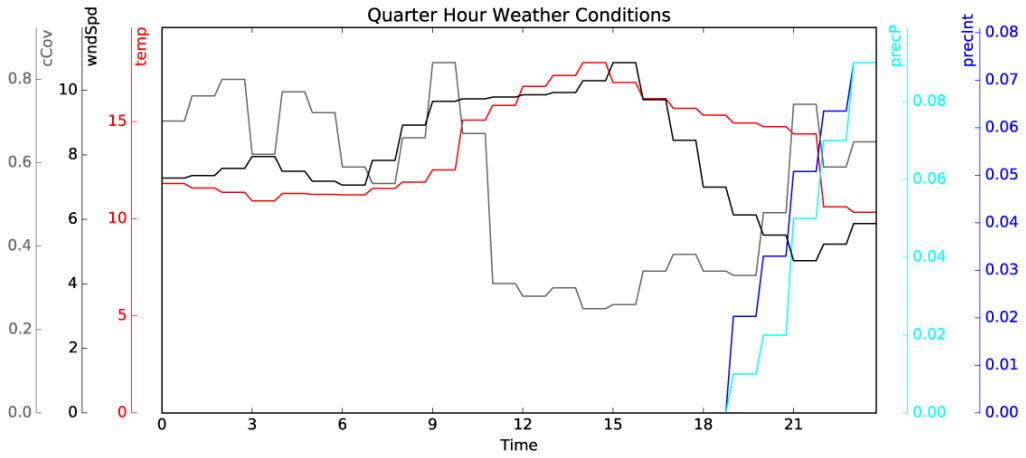


Figure 8.7: Quarter Hour Weather Counts for a sample day (27/10/2015).

[1]For a comprehensive list, see the darksky.net documentation: https://darksky.net/dev/docs/time-machine

The finest resolution offered by this service was at hourly blocks, and these values were assumed to apply for the whole hour. This resolution captures significant variation throughout the day, as can be seen in Figure 8.7. After collection, weather data was disaggregated to 15-minute intervals, and collated with the SocM data collected, after removing days where SocM collection had failed. The final quarter-hour weather conditions were used in the *Forecast Model* discussed in section 8.3 as the independent variables, in order to predict quarter-hour SocM values.

8.2.2 Activity Site Surveys

Actual visitor activity data was collected during four site visits, with the specific goal of recording the locations and activities of actual visitors at the park, so as to have baseline *'ground truth'* information on activity throughout the park. The method used for recording visitor activity on-site has been presented in detail in a previous chapter (*Section 6.3*), however a brief summary will be offered here.

The site surveys were conducted using a fieldwork application for mobile devices, which records the GPS location of the mobile device along with other relevant data (date, time, etc., as well as customized data options) when the user presses a button in the application interface. Using the app, the surveyor walked along a predetermined path, triggering a new record in the app every time they encountered a park visitor within a certain radius (approximately 100 meters). While planning the path, an attempt was made to strike a balance between area coverage and traversal duration, so that walking along the path covered as much of the area as possible within a reasonable time. Due to the design of the fieldwork app itself, where a new event records the device's (i.e. surveyor's) exact location instead of the location of the park visitor being recorded, the datasets were subsequently imported into GIS software, and data points were given new positions, calculated to be within 100 meters of their recorded location. These redistributed locations were then used as the final working locations.

The four surveys took place on different days in October 2015. The dates were

divided into two pairs, so that each pair contained dates on the same week, and so that each pair contained one weekday and one weekend day, specifically Sunday. Therefore, two Sundays and two weekdays were covered in total, and furthermore the two survey date pairs happened a week apart. Finally, all surveys took place at approximately the same time of day, in early to mid afternoon. A survey summary is presented in Table 8.3.

Set	Date	Day	Start Time	End Time	Duration (mins)	Total Visitor Count
1	02/10/2015	Friday	13:11:40	14:49:26	97	2687
	04/10/2015	Sunday	14:04:54	15:42:33	97	5424
2	11/10/2015	Sunday	14:13:57	15:58:11	104	4340
	14/10/2015	Wednesday	13:13:01	14:39:26	86	1662

Table 8.3: HYP Site Survey Summary

As can be seen in the summary, there is a definite tendency for an increase in visitors on Sundays, with larger visitor numbers in weekends compared to weekdays. This is expected, as more people would visit the park for leisure activities on a weekend day, however it is interesting to note the scale of increase, as visitor numbers more than double on weekends. In addition to recording total visitor numbers, park user activities were captured in two categories, moving or stationary activities. A summary of user activities can be seen in Table 8.4. A note on capturing moving visitors: As was explained in section 6.3, only visitors that crossed paths with the surveyor were captured (i.e. walking in opposite directions), and therefore it was assumed that approximately half of the moving visitors were recorded. This value is represented in column 'Walking Counted', in column 'Walking Estimated' the value is doubled and used as final estimate. There appears to be a semi-fixed relationship between moving and stationary park users, with the numbers being roughly similar and a ratio (sitting/walking) of approximately 1. Furthermore, S/W ratio seems to be higher on Sundays compared to weekdays, although this might be explained through the analysis of the spatial distribution of activities that follows.

Considering the spatial distribution of activity throughout the park, the following figures demonstrate the workflow necessary to produce the final dispersed activity

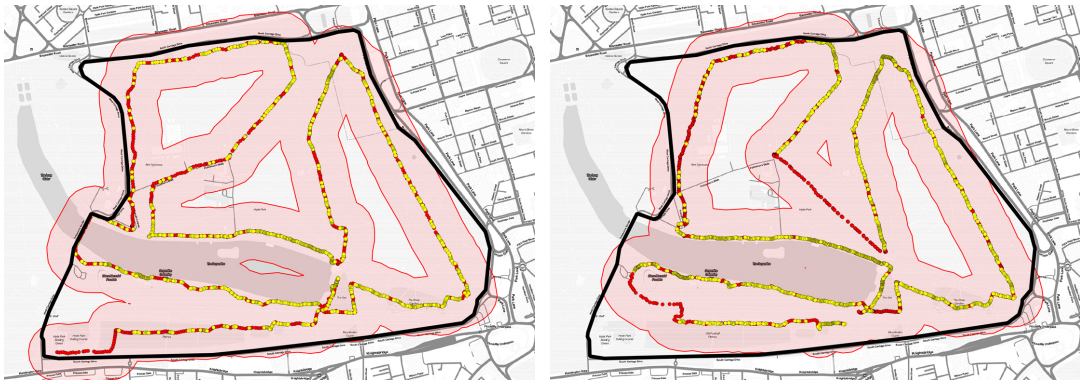
Set	Date	Day	Total Visitor Count	Sitting Counted	Walking Counted	Walking Estimated	Sitting Percentage
1	02/10/2015	Friday	2687	1693	994	1988	45.99%
	04/10/2015	Sunday	5424	3904	1520	3040	56.22%
2	11/10/2015	Sunday	4340	3082	1258	2516	55.06%
	14/10/2015	Wednesday	1662	1150	512	1024	52.90%

Table 8.4: HYP Site Survey Visitor Statistics

maps. Figure 8.8 shows the survey paths, along with area coverage buffers. Of the two large undocumented areas (as seen in images 8.8b, 8.8c, and 8.8d), the one in the north-west quadrant of the park is covered in meadows and clumps of trees in addition to containing two large walled-off areas (the Royal Parks Nursery, and the Ranger's Lodge and Old Police House), while the strip in the east side comprises of large open lawns. Activity in both these areas was found to be similar to the surveyed areas around them (little to no activity in the meadow, small-to-fair amount of activity consistently evenly distributed throughout the lawns). Figure 8.9 shows the final locations of activities, after points were repositioned within the 100m buffer area. Activity heatmaps of the finalized locations are shown in Figure 8.10.

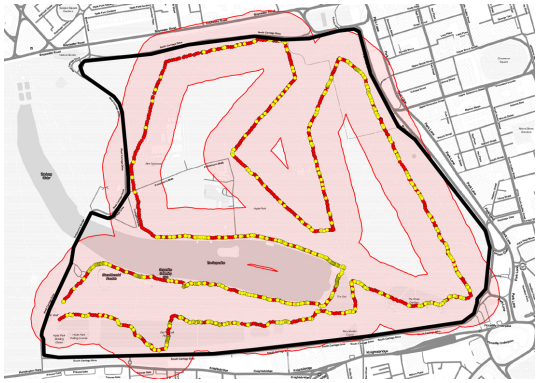
The heatmaps were generated using a straightforward feature count for each cell, with query distance set at 100 meters, as identified in literature to be the upper distance limit in human interaction (Hall, 1966a, Ciolek, 1983, Gehl, 1987). The heatmaps represent crowd densities measured as people per 3.14 hectares (due to the 100m radius), and may be further interpreted as mapping the *'perceived vitality'* at each cell. For Hyde Park, a number of interesting points in terms of recorded activity have been identified, and their locations have been highlighted in green in image 8.10a.

Location 1, found at the north-east corner of the park, near the entrance gate and lawns, is consistently showing some activity in all surveys. However, it is evident that it becomes a significant hotspot of activity on Sundays. This is due to the fact that the *Speakers' Corner* is located there, an area used by members of the public for public speaking, debates, and discussions. Although the Corner is open to everyone anytime during park operation hours, orators at the Corner tend to draw

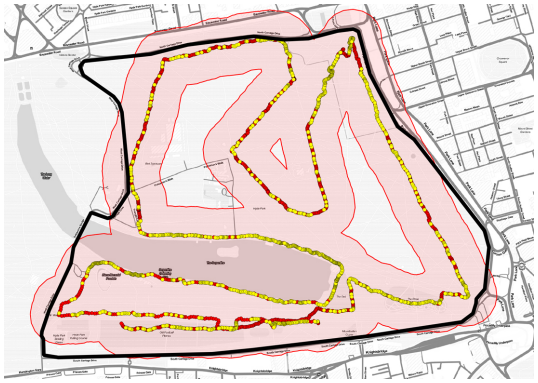


(a) Set 1 Weekday

(b) Set 1 Sunday



(c) Set 2 Weekday



(d) Set 2 Sunday

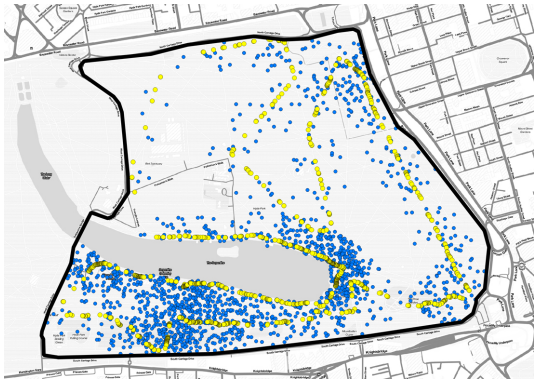
Figure 8.8: Hyde Park Site Surveys Paths. The images are arranged by set in rows, by day type in columns (weekdays in column 1, Sundays in column 2). The survey path is shown in red points. Yellow points mark the locations of individual records. The red offset around the path highlights the surveyed area (100 meters around the path).

large crowds on Sundays, as is evident in the heatmaps, which might further explain why stationary activities are relatively increased on Sundays. In addition to the Speakers' Corner, the north-east area opens into the expansive lawns which tend to have groups of people sitting, and furthermore the Marble Arch Tube station is located right outside the north-east corner of Hyde Park. As such, activity in this location is consistently above average at least.

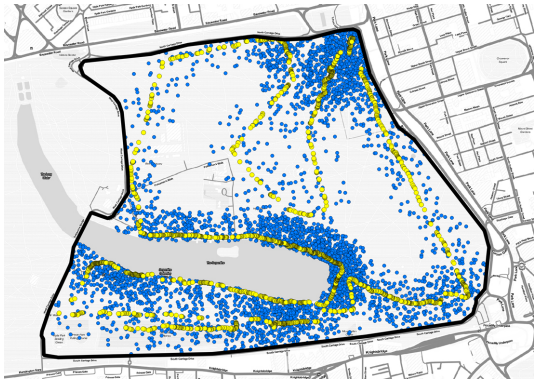
Location 2 marks the south-east end of the body of water known as *The Serpentine*. The waterfront at this location along with the area immediately to the east are consistently shown to be the largest hotspot of activity in the park, across all four surveys. Multiple potential reasons for this increased activity have been iden-

(a) Set 1 Weekday

(b) Set 1 Sunday



(c) Set 2 Weekday

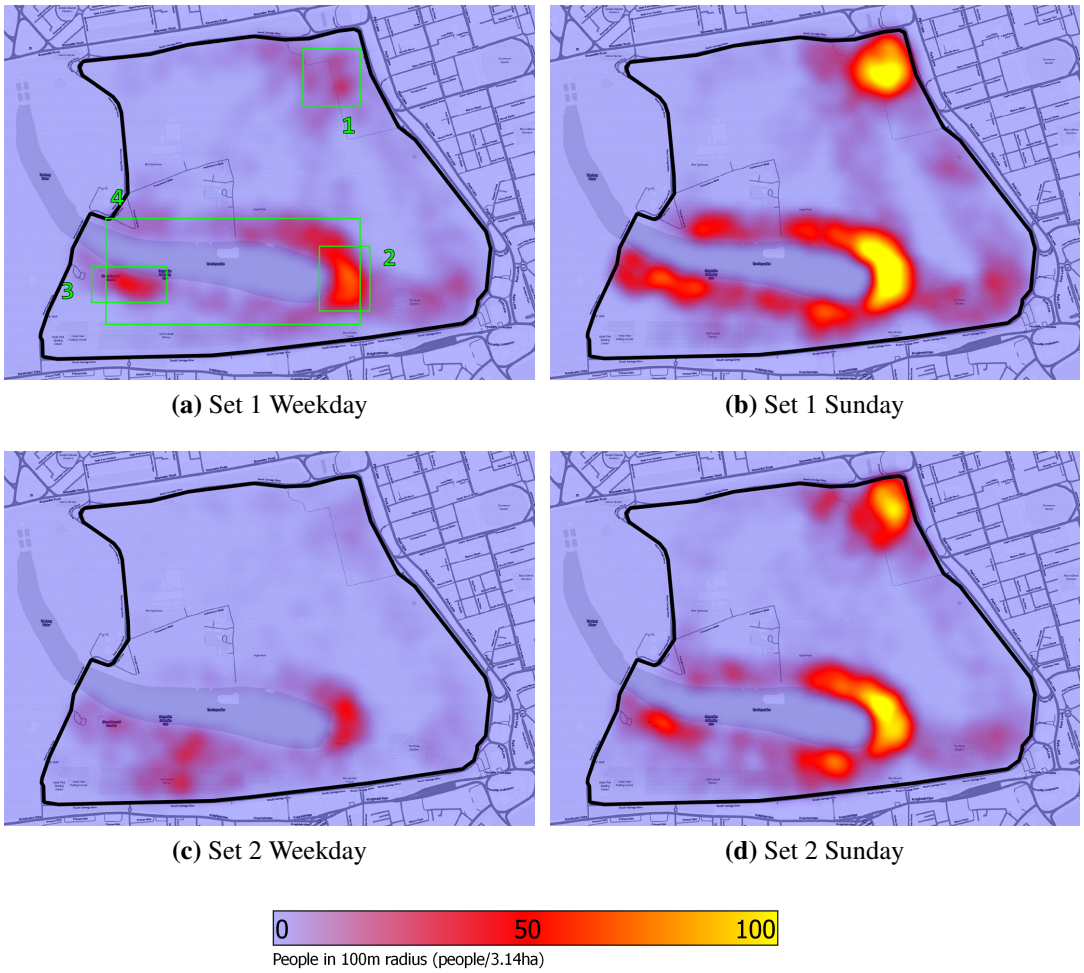


(d) Set 2 Sunday

Figure 8.9: Hyde Park Site Survey Point Recalculation. Yellow points mark the recorded locations, blue points mark the recalculated locations.

tified. First, the Serpentine runs uninterrupted through most of Hyde Park, dividing it into two parts, and location 2 is the westernmost point at which Hyde Park can be traversed north to south. Therefore, it constitutes a natural bottleneck regarding visitor flows, which adds to the perceived crowdedness. Second, a restaurant with plenty of outdoor seating operates at this location, which significantly adds to the observed stationary activities. These two factors combined might help explain the consistently high observed activity at this location.

Location 3 is found in the south-west of the park, on the south waterfront of the Serpentine. It appears as a low to medium intensity hotspot in most surveys (8.10a, 8.10b, 8.10d). Two features are found here that seem to attract activity, one being a medium-size cafe with outdoor seating on the waterfront, the other being the *Diana Memorial Fountain*, a large landscaped water feature with informal seating areas



(a) Set 1 Weekday

(b) Set 1 Sunday

(c) Set 2 Weekday

(d) Set 2 Sunday

Figure 8.10: Hyde Park Site Survey Activity Heatmaps.

and a lawn. Both of these features appear to consistently attract seating activities, and they may be potentially supporting one another, with visitors moving between the two.

Location 4 constitutes the waterfront of the Serpentine, which has consistently been observed to gather medium intensity activity along its paths. The paths have been designed as a promenade around the Serpentine, which along with it being the only large body of water might explain its attractiveness for walks around the water.

8.3 Forecast Model

The previous section presented all relevant data that has been collected for this case study. It highlighted issues with some of the datasets, and explained the various reasons why some potential datasets were eventually dropped from the case study. This section will present how these datasets were used to form and calibrate a *Real-Time Public Space Activity Forecast Model* for Hyde Park, so that total visitor volumes could be continuously estimated. As was explained in *section 5.2: Forecast Submodel*, from the two forecasting approaches considered (*subsection 5.2.1: Visitor Supply Approach* and *subsection 5.2.2: Total Visitor Volume Approach*), only the *Total Visitor Volume Approach* proved to be viable. The rest of this chapter will continue with the implementation of this approach.

Some initial observations have already been made regarding apparent relationships between SocM and weather and temporal conditions, which will be used as the starting point for the rest of this section. Section 6.1 established that some correlation exists between SocM and weather/temporal conditions, at multiple temporal resolutions. At a daily level, it has been shown (Figures 6.8a, 6.8b) that SocM shows little to no correlation with temperature variation; however daily temperature range seems to correlate with SocM variance (6.8c), thus hinting at other environmental factors having an effect on SocM activity (Easterling et al., 1997). Precipitation probability and intensity do exhibit some relationship to SocM (Figures 6.8d, 6.8e), although not any particular linear correlation. Finally, cloud coverage and wind speed both indicate a negative linear correlation with SocM (Figures 6.8f, 6.8g). At hourly level, scatter plots of SocM against environmental conditions show a positive correlation *within individual days* but no overall pattern over the whole dataset, as can be seen in Figure 8.11 (Other weather conditions provided similar results). However, at this temporal resolution, it appears that time-of-day is a major driving factor, as can be seen in Figure 6.11.

Given the above observations, it is assumed that some relationship exists between weather and temporal parameters and SocM activity, and it is further hypothesized

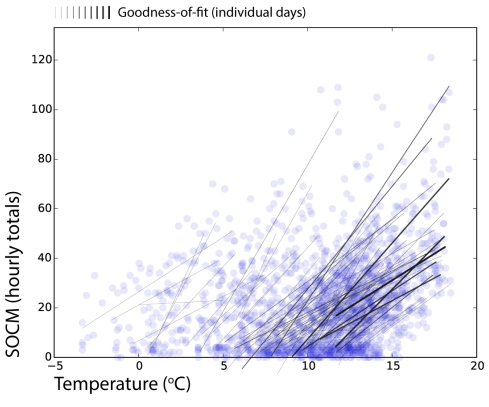


Figure 8.11: SOCM vs. Temperature - Hourly

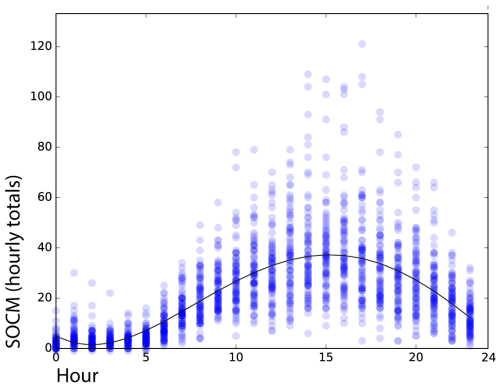


Figure 6.11: SOCM vs. Hour (repeated from page 158)

that SocM activity correlates with actual activity. The aggregate activity forecast model will be built around these two hypotheses.

8.3.1 Model Formulation

The abstract forecast model of current activity from *subsection 5.2.2: Total Visitor Volume Approach* has been defined as

$$P_t = T_t * W_t * p + A_t + e$$

where P_t is the total population of park visitors active in the area at time t, T_t and W_t are time and weather modifiers respectively at time t, p is a population coefficient, A_t stands for any special attractors in the area at time t, and e is a constant. A_t was considered to be set using planned events at the park, captured using Facebook's API. As that data source was dropped from this case study, the A_t term was also removed from the formula, as there was no reliable way to capture events and attractions. Therefore the overall forecast model is now defined as

$$P_t = (T_t * W_t) * p + e$$

Furthermore, given the observed relationship between SocM and weather/temporal conditions, it is assumed that a secondary model exists, where at time t, SocM events $SocM_t$ can be estimated as a function of weather and temporal parameters, so that

$$SocM_t = T_t * W_t$$

and therefore

$$P_t = p * SocM_t + e$$

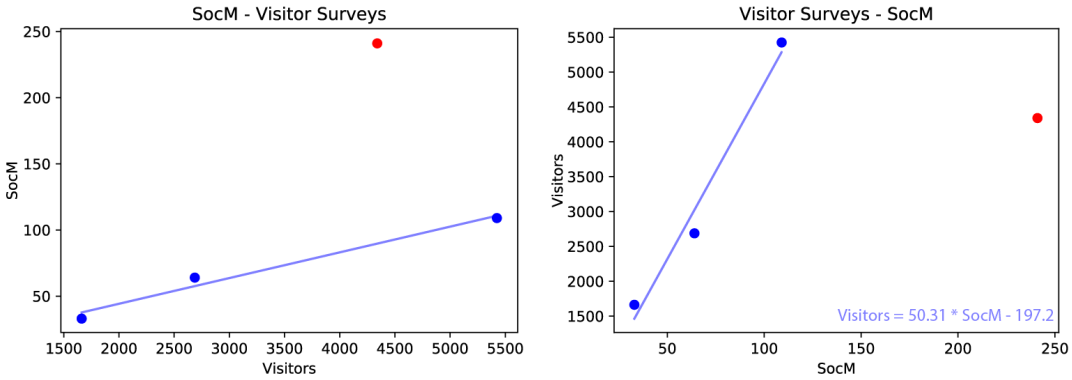
Data is available for SocM and visitor population during the survey periods, by counting all SocM events originating from Hyde Park in the duration of a survey, and a summary can be seen in Table 8.5. Set 2 Sunday shows an exceptionally high SocM count compared to the other three survey dates. This is due to a sports event (half marathon) that took place on that day before the site survey, and was evidently discussed in social media. Disregarding this value, the remaining three seem to be consistent at under 50 people per SocM event.

Set	Date	Day	Total Visitor Count	SocM	Ratio (V/S)
1	02/10/2015	Friday	2687	64	41.984
	04/10/2015	Sunday	5424	109	49.761
2	11/10/2015	Sunday	4340	241	18.008
	14/10/2015	Wednesday	1662	33	50.363

Table 8.5: Site Survey - SocM Summary

Using the information above, it is possible to calculate p and e values for the forecast model. As can be seen in Figure 8.13a, a positive linear relationship exists between actual visitor activity and SocM activity. Running a linear regression for visitor population by SocM, as shown in Figure 8.13b, results in the equation $V = 50.31 * S - 197.2$, and therefore $p = 50.31$, and $e = -197.2$.

Having established a connection between SocM and visitor numbers, the next step requires the equation connecting SocM and weather and temporal parameters to be



(a) Socm - Visitor Population **(b)** Visitor Population - SocM

Figure 8.13: Visitor Population - SocM Correlation

identified and defined. In its abstract form it has been defined as

$$SocM_t = T_t * W_t$$

so that at any time t, the total number of SocM events in the park can be calculated as a function of weather conditions and time. This SocM-Weather/Time model essentially establishes the real-time nature and forecasting capabilities of the overall model, as it ties SocM activity first to weather conditions, which can be reliably forecast for short periods in the near-future, and second to time, which is a known variable. It was decided that some form of a multiple linear regression model would be most suitable for this model, using various weather and temporal parameters as the independent variables, and SocM as the dependent variable.

8.3.2 Model Calibration

For the calibration of the multiple linear regression model, the SocM-weather data was used, which as discussed earlier was disaggregated to 15 minute intervals, and SocM records were summed for the past 6 time steps. The aim of the calibration was first to identify the form of the linear model, identifying dominant independent variables and their relationships, and second to set the model coefficients. For this purpose, multiple iterations of linear regressions were implemented using different

combinations of the independent variables, and the adjusted R-squared values were compared between implementations to determine goodness of fit.

Some initial considerations were made regarding some of the dataset's properties, specifically the different day types. There is significant variation in daily SocM totals between different day types, most obvious between Sundays and weekdays. Originally the aim was to include day type as an additional variable in the linear model. However, as has been observed already, time of day appears to be the dominant variable that drives SocM changes throughout the day. If day type was to be included as an additional variable, this would assume that the activity trendline throughout the day would follow a similar curve between Sundays and weekdays (peaking at the same times, etc.). To avoid including this assumption in the model, it was decided to split the dataset for different day types using five categories: 'Week' (all seven days of the week as one category), 'Weekdays' (Mondays-Fridays as one category), 'Weekends' (Saturdays-Sundays as one category), 'Saturdays', and 'Sundays', and identify the most applicable classification through the calibration process.

Furthermore, it has been established that for smaller time steps (less than an hour), SocM activity throughout the day is best approximated using a polynomial function of time. For the calibration process, polynomials at multiple degrees were considered for different implementations, specifically 3rd, 4th, and 5th degree polynomials, in order to identify the most applicable. These were further combined with weather parameters in order to test best fit. Calibration runs were performed in stages, at each stage refining the parameter set. First the polynomial degree was established, second different weather parameters were examined for best fit, third variable combination mode was tested (additive or multiplicative) , and fourth the combination of multiple weather parameters was examined. For each run, the adjusted R squared values were recorded, and later compared between runs to determine best model fits. A summary of fit statistics is presented in Table 8.6.

As can be seen both in the table as well as in Figure 8.14, 5th degree polynomial

Model	Week	Weekdays	Weekends	Saturdays	Sundays
SocM : hr^3	0.5162	0.5719	0.5754	0.5583	0.6397
SocM : hr^4	0.5180	0.5740	0.5772	0.5625	0.6403
SocM : hr^5	**0.5204**	**0.5743**	**0.5872**	**0.5744**	**0.6497**
SocM : hr^5 + temp	0.5282	0.5893	0.5875	0.5772	0.6496
SocM : hr^5 + cCov	**0.5380**	0.5874	**0.6167**	0.6018	**0.6870**
SocM : hr^5 + wndSpd	0.5377	**0.5902**	0.6132	**0.6125**	0.6752
SocM : hr^5 + precP	0.5317	0.5792	0.6106	0.5997	0.6580
SocM : hr^5 + precInt	0.5288	0.5790	0.6010	0.5870	0.6567
SocM : hr^5 * temp	0.5301	0.5928	0.5884	0.5770	0.6519
SocM : hr^5 * cCov	**0.5542**	**0.5998**	**0.6445**	**0.6361**	**0.7206**
SocM : hr^5 * wndSpd	0.5445	0.5948	0.6281	0.6280	0.6929
SocM : hr^5 * precP	0.5427	0.5841	0.6365	0.6312	0.6675
SocM : hr^5 * precInt	0.5393	0.5850	0.6329	0.6279	0.6696
SocM : hr^5 * cCov * temp	0.5606	0.6145	0.6502	0.6376	0.7283
SocM : hr^5 * cCov * wndSpd	**0.5680**	**0.6154**	0.6565	**0.6538**	0.7391
SocM : hr^5 * cCov * precP	0.5671	0.6060	0.6703	0.6497	0.7505
SocM : hr^5 * cCov * precInt	0.5661	0.6079	**0.6707**	0.6498	**0.7543**

Table 8.6: Adjusted R^2 for SocM - Time/Weather Linear Model by Coefficient. Model best fits for each calibration stage and day type are highlighted in bold.

curves presented the best fit. Regarding weather parameters, it appears that best results were generated by using either cloud coverage or wind speed as a parameter, depending on day type classification. However, multiplicative combination of terms resulted in significantly increased results using cloud coverage as the single weather parameter. Finally, although adding additional weather parameters as variables increased R^2 scores, weather parameters other than wind speed seemed to have a greater effect.

Ultimately, it was decided that a smaller parameter set was preferable to increased scores, and therefore 5th degree polynomials combined with cloud coverage were chosen as the most applicable model fit. Regarding day type classification, it was decided that three different classes would be used: Weekdays, Saturdays, and Sundays. As can be seen in Figure 8.15, a single class for the whole week provided the poorest results, and therefore splitting into multiple cases was required. Given similar values between Saturdays and Weekends, it was decided that splitting the

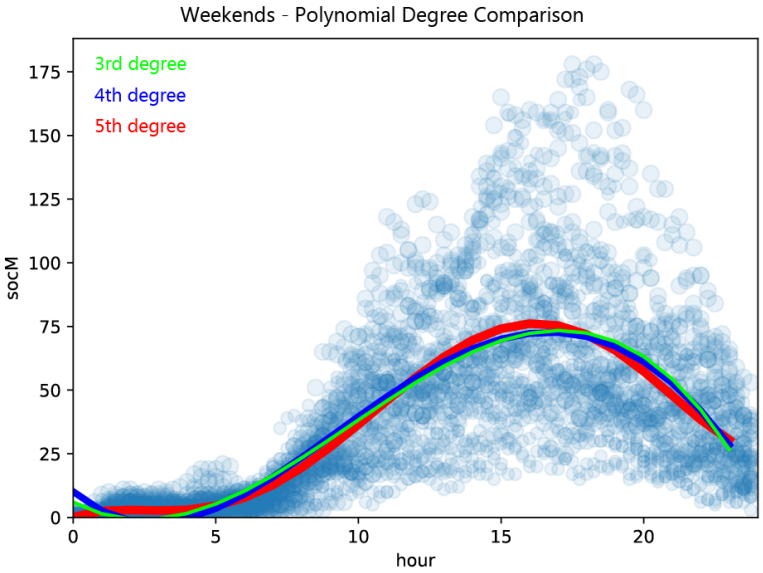


Figure 8.14: Polynomial Degree Curve Fit Comparison

Weekend class further into Saturdays and Sundays was required, as it seems that the two day types exhibit different activity patterns.

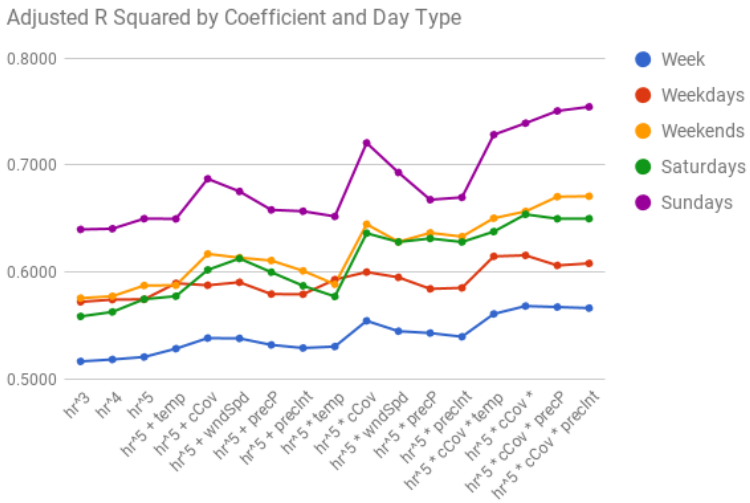


Figure 8.15: Adjusted R^2 by Coefficient and Day Type

8.4 Spatial Disaggregation Model

Having established a short-term predictive model of total visitor activity in Hyde Park, the next part of the model requires the disaggregation of total activity val-

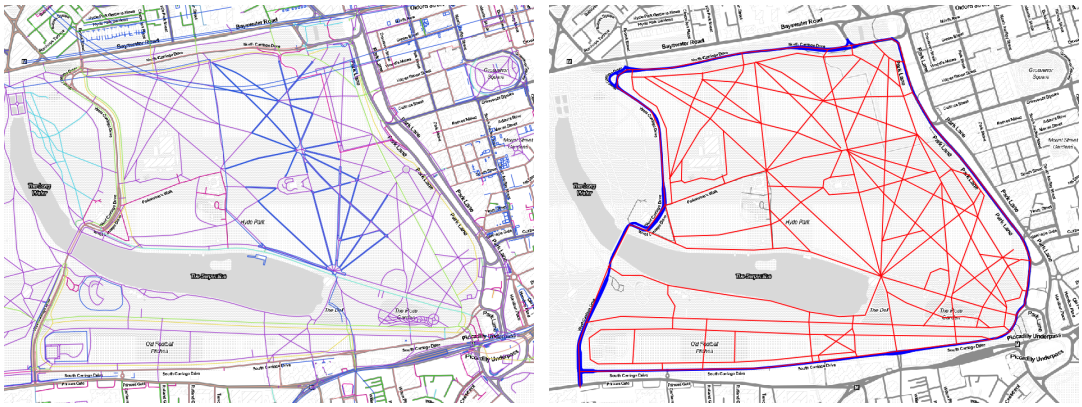
ues into individual visitors, accurately dispersed in the area of interest. This was achieved via an Agent-Based Model (ABM) of Public Space Activity (PSA), calibrated to park visitor activity. The overall framework of the ABM used for this SDM has been discussed in detail in *chapter 7: Modelling Spatial Behaviour*. This section will discuss matters relating to this ABM, as they were approached through this case study. The section will focus on two main points: First, the generation of the model environment (a 3D virtual representation of Hyde Park), and second and most important, the setup and calibration process of the ABM itself.

8.4.1 Virtual Environment Generation

The importance of spatial three-dimensional models has already been established for the context of this work. Although at first the necessity of the vertical dimension was contemplated, it was decided that the Hyde Park case study would be implemented in a 3D environment. Some notes on this decision: At first glance, Hyde Park does not offer any strong arguments for the use of the third dimension; it lacks any additional levels either under or over the ground level, with all activity taking place on a single surface, and furthermore, its topography and landscape do not exhibit any drastic differences, with maximum height differences in the range of 60 meters across lengths of approximately 1 km, rendering them effectively negligible. It would therefore be definitely acceptable to develop the ABM in a two-dimensional environment. However, a two-dimensional implementation was considered to be a potential limitation, as any further application to additional areas would require significant expanding of the methodologies developed in this first case study, should they present more complex topography. Given that this was the first case study, it was decided to take an approach similar in spirit to the use of only publicly available datasets and approach this from a broader perspective so as to allow for additional applications, and therefore develop this model in three dimensions.

A primary requirement then for this case study was for the ABM to run in a 3D en-

vironment. At the time when the case study was developed, no existing 3D model of Hyde Park could be found at an acceptable resolution. Therefore, a new 3D model of Hyde Park was created, using data from the UK Ordnance Survey and OpenStreetMap (OSM). Data was cleaned, manipulated, and modified using a series of software: QGIS, Esri CityEngine, Autodesk 3DS Max, and ultimately Unity, where the ABM was implemented. Specifically, the datasets used were the OS *MasterMap topography layer* and *Terrain 5 DTM* (Ordnance Survey Digimap Licence), and the OSM geodatabase (©OpenStreetMap contributors).

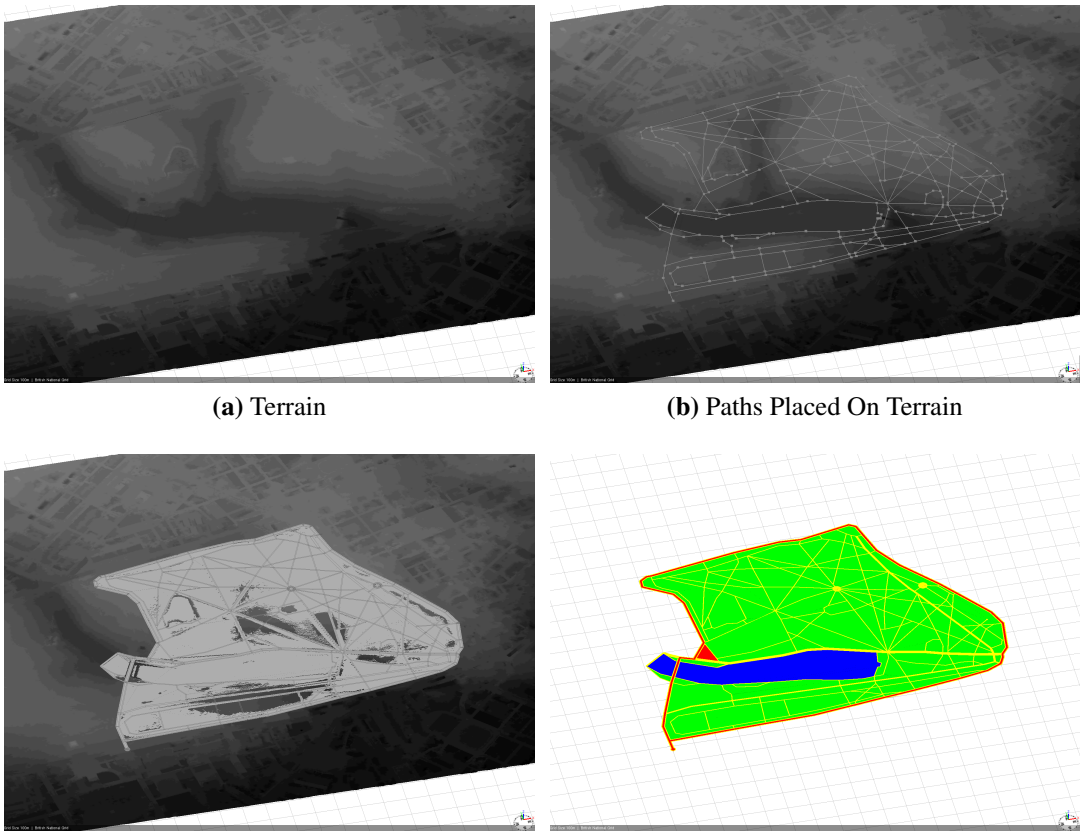


(a) All OSM Path Types, Grouped by Tag **(b)** Simplified Path Geometries

Figure 8.16: OpenStreetMap Path Geometry Cleanup

The OSM geodatabase was used as the basis for the generation of the 3D environment. The first task was to create path geometries for all pedestrian paths in Hyde Park. For this task, polyline geometry was imported into QGIS, classified by type obtained by OSM tags, as can be seen in 8.16a, and pedestrian paths were isolated. Pedestrian paths included polylines with the tags *'cycleway'*, *'footway'*, *'path'*, *'pedestrian'*, and *'service'*. The isolated geometries were simplified and merged into one category. The resulting paths are seen in 8.16b. The second task was to create a list of all tree locations in Hyde Park, so that trees could be placed in the 3D model. Information on tree locations exists in two forms in OSM: as point geometry for individual trees, and as polygon geometry outlining wood areas, and both types exist in OSM data of Hyde Park. Tree densities were calculated from existing tree point geometries, and the wood polygon areas were filled with additional

tree points according to average tree density for the park. The code used and overall tree reconstruction process is detailed in section A.5.



(a) Terrain **(b)** Paths Placed On Terrain

(c) Shapes from Paths **(d)** Meshes from Shapes

Figure 8.17: Generation of 3D geometry from shapefiles in Esri CityEngine

The next step in the generation of the 3D environment required the conversion of geodatabase information into actual 3D mesh geometries. This was performed in Esri CityEngine, a software specializing in procedural 3D modelling. First, the terrain topography was recreated (8.17a), using a Digital Terrain Model (DTM) from Ordnance Survey. The simplified path geometry that was created previously in QGIS was then imported and nodes were placed on the terrain (8.17b). Third, using CityEngine's procedural rules, path lines were expanded into shapes with appropriate widths, and the areas in-between paths were filled with solid polygons (8.17c). Finally, the resulting shapes were converted into 3D geometries, given 'terrain-type' properties, and grouped by type (four terrain types were defined, *'path', 'green', 'water',* and *'road'*, 8.17d). The resulting geometry was exported as an FBX file,

and further refined in Autodesk 3DS Max, before imported into Unity.

The final step in the environment generation process required all assets to be imported into Unity, the platform where the ABM was developed in. Unity supports 3D geometries, and so the FBX file was added to the Unity project without further work. Tree locations were imported as a coordinate list in text format, parsed, and placed in the correct locations. Gates and points of interest/features were added manually for the Hyde Park model. The combined 3D model is shown in Figure 8.18.

8.4.2 Model Calibration

The park visitor ABM rules and parameters implemented in this model have been discussed extensively in Chapter: 7. This section will present the process of calibrating and verifying the model. As a general rule, agent behavioural parameters were set to reflect observations of human behaviour in public spaces, as discussed in *chapter 2: Understanding Public Space Use*, and model calibration involved tweaking and refining parameter values. Unfortunately, no publicly available dataset was found at a high enough spatial and temporal resolution to calibrate against, and therefore the calibration process was done against park visitor data captured during site surveys. Specifically, survey dates 04/10/2015 and 14/10/2015 were used for ABM calibration.

No aggregate activity forecast data was used in the calibration runs to control the population. Instead, the model was set to initialize with a visitor population of 800, set to increase by 200 at each update, and stabilize at 1600. After a few updates at a population of 1600 and park activity had stabilized, relevant model output was captured. For visual verification of calibration process, a grid of cubes was set up over the environment area, at 100 meters, with every cube updating its appearance (height and opacity) depending on the number of agents within the grid square.

The first point in the calibration process was to establish gate importance, whether

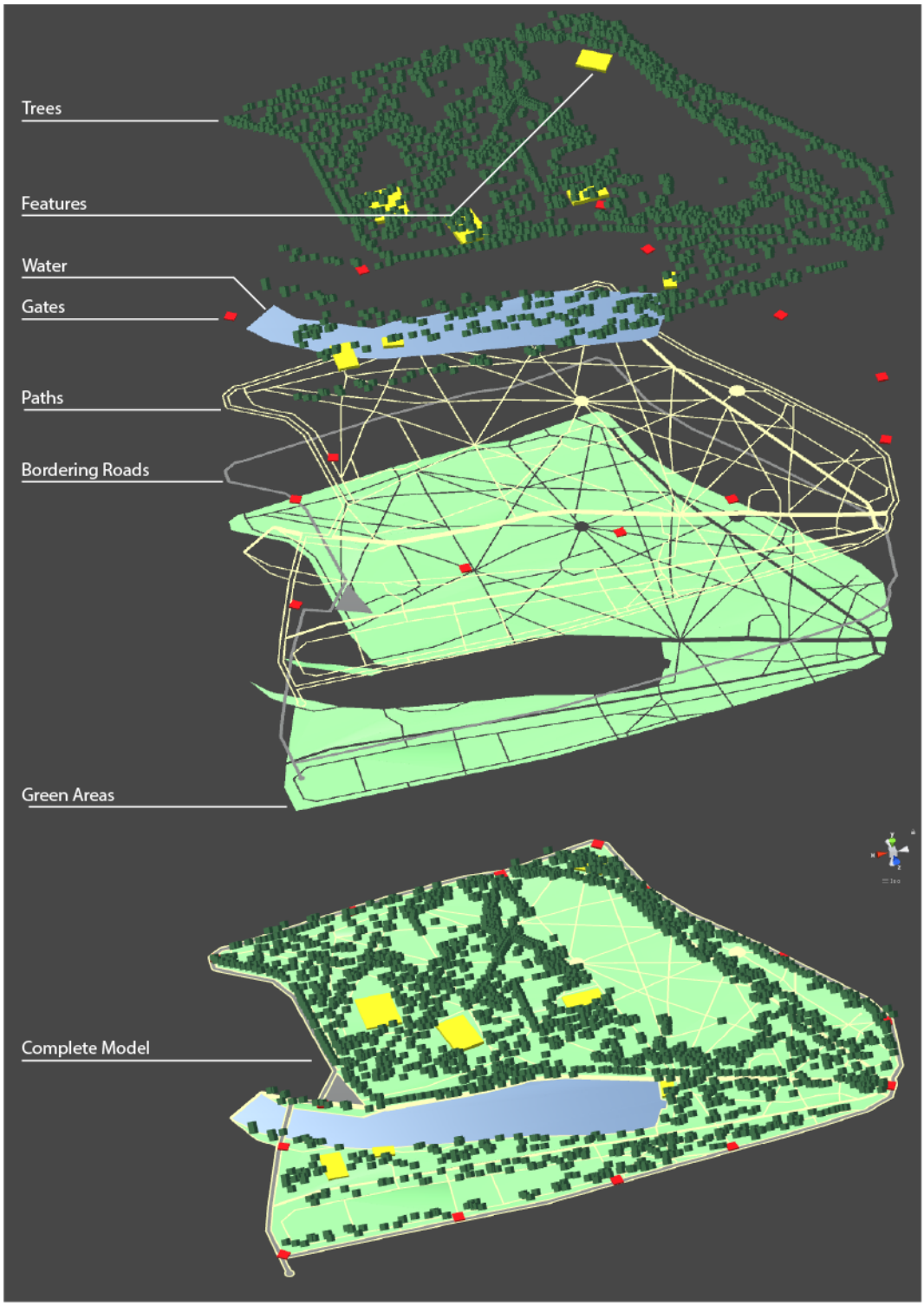
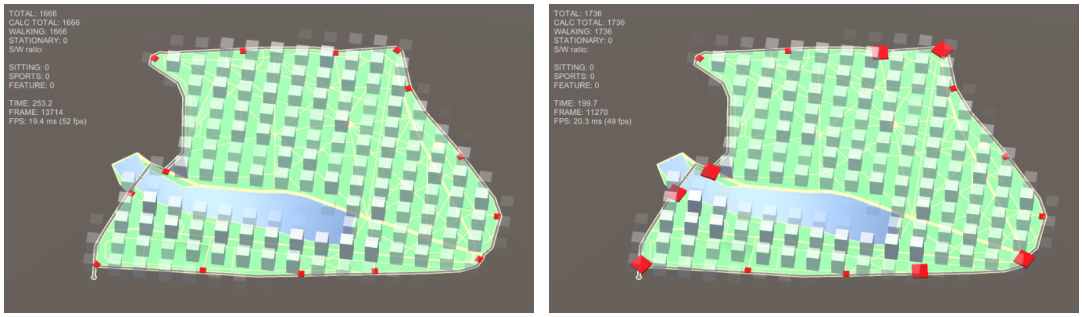


Figure 8.18: Hyde Park 3D Environment in Unity

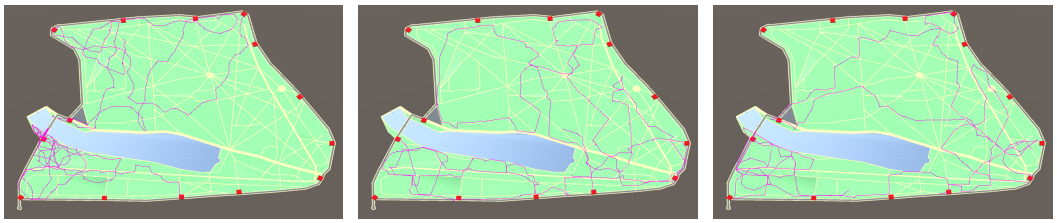
gate preference by agents has any effect on overall model behaviour, and if so, determine gate weight values for each gate. No dataset was available to calibrate gate importance, and so an exploratory comparison test was performed, first with all gates having equal weights, then with some gates having a weight of 5, meaning they were 5 times more likely to be chosen by an agent to enter or exit the simulation. The results of both runs are seen in Figure 8.19. As can be seen in the figure, gate preference does not seem to have a large effect on overall agent distribution.



(a) No Gate Preference **(b)** With Gate Preference

Figure 8.19: Gate Preference Comparison. Larger blocks in 8.19b mark gates with increased weights.

Another model parameter examined here was the agent view distance. This distance was originally set at 100 meters based on literature, however as can be seen in 8.20a, this resulted in agents tending to move in circles in small areas, and getting tangled in slightly more complex geometry. A view distance of 200 meters was used, as a balance between expanding movement over the whole area and keeping the value close to observed values in literature.

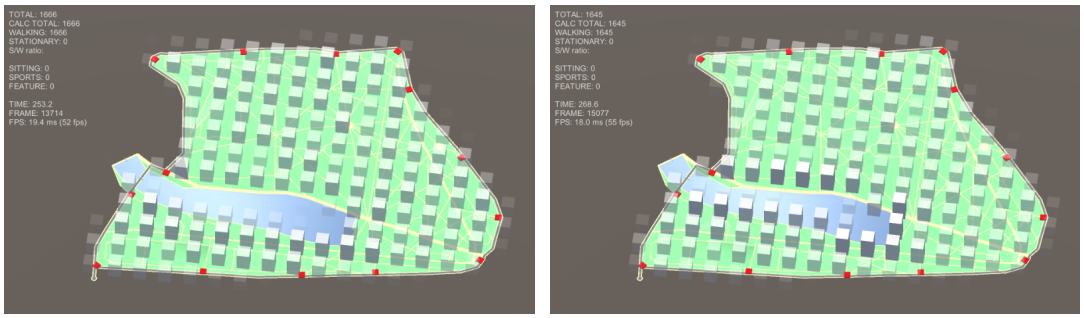


(a) Agent View Distance 100m **(b)** Agent View Distance 200m **(c)** Agent View Distance 300m

Figure 8.20: Agent View Distance Comparison. Images show the paths of two agents in each simulation run after 4500 frames.

Another matter of consideration regarding view distance and movement was

whether a next path length limit should be set, when compared to the straight-line distance between origin and destination. A distance cap was placed at 2 times the straight-line distance, meaning that a new random target would only be considered valid if the path to it was at most twice as long as the direct distance to it. The effect of this limit can be seen in Figure 8.21. In 8.21b, where no path length limiting is applied, agents tend to circumnavigate the Serpentine River more often. Ultimately it was decided to not place this limit, under the assumption that as long as a location was within effective view, it would be considered valid.



(a) With Path Length Limit **(b)** Without Path Length Limit

Figure 8.21: Path Distance Limiting Comparison

Agent group size was drawn from a table of integer values between 1-4, with fixed probabilities as discussed in subsubsection 7.3.3.3. Data on visitor statistics was gathered from visitor surveys that took place in Hyde Park (Ipsos Mori, 2015a), where visitor group average size was found to be approximately 1.72. Average simulation group size over time is shown in Figure 8.22, converging to 1.72 over the course of the simulation.

Agent lifetime, signifying visit duration, was similarly drawn from a distribution based on visitor surveys (Ipsos Mori, 2015a), and set individually at agent creation. Average lifetime over time can be seen in Figure 8.23. The model seems to be underperforming slightly, stabilizing at an average agent lifetime of approximately 4500 ticks (75 minutes), compared to visitor average 4842 (80.7 minutes). This is possibly due to the model population control routine, which allows agents to be removed from the simulation prematurely, and therefore introducing a slight bias.

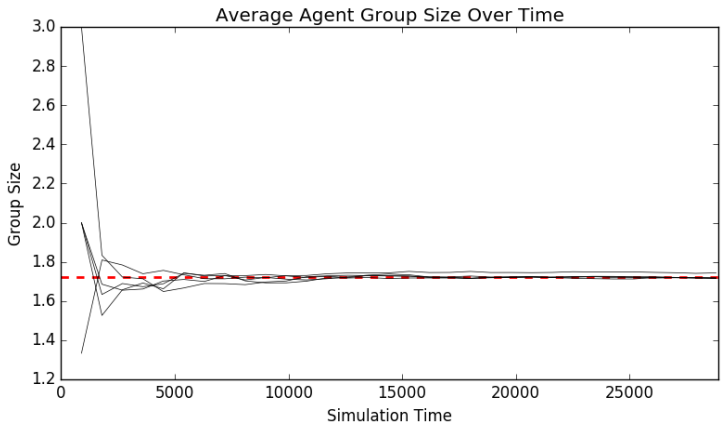


Figure 8.22: Agent Group Size Verification

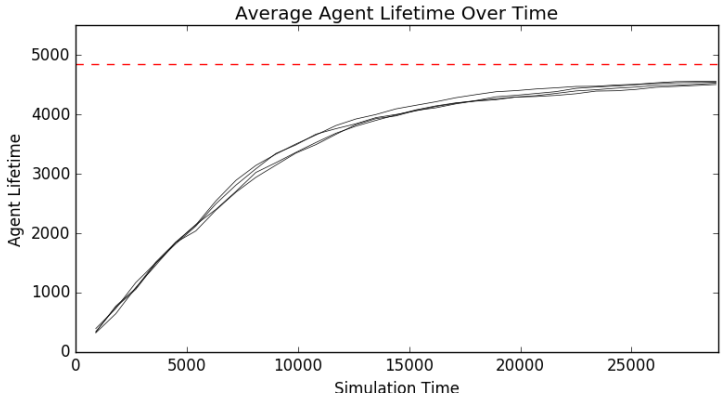


Figure 8.23: Average Agent Lifetime

8.5 Evaluation

The overall Real-Time, Public Space Activity model of Hyde Park has been presented thoroughly at this point. It consists of two sub-models, the aggregate activity forecast model, and the spatial disaggregation model. The overall model uses real-time Social Media (SocM) and weather data as input, and generates a simulation and visualisation of activity in the park in real-time, at the individual visitor scale. Overall model performance will be evaluated in this section.

Some notes on validation datasets: The objective of the overall real-time activity model was to simulate individual visitor activity for any point in time, i.e. the model aims to perform at high temporal and spatial resolutions. A dataset capable of validating the overall model would therefore need to have the same characteristics,

however no such dataset was available. In light of this, it was decided to evaluate sub-model performance separately: temporal accuracy was evaluated through a validation of the forecast sub-model, and spatial accuracy through a validation of the spatial disaggregation model.

The aggregate activity forecast sub-model was evaluated against a subset of the SocM-weather dataset, which included available dates in March 2016, 28 days in total. SocM values were calculated for each quarter-hour period as a sum of the past 90 minutes, and compared to observed data. A range of error measures was captured: mean absolute error (MAE) and root mean squared error (RMSE) as quick scale-dependent error metrics, mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (sMAPE) as percentage error metrics, for comparison across the dataset. The full validation set is presented in Table 8.7.

The four metrics were consistent in identifying the most (17/03/2016) and least (25/03/2016) accurate forecasts, and furthermore were broadly consistent in recording overall model error. Figure 8.24 shows two examples of the two outliers, and a full list with validation graphs for all dates is included in Appendix C.1. As can be seen from the graphs, the forecast model developed here performs well at capturing and predicting overall activity for 'default' days, i.e. days with standard conditions, and can therefore adequately provide an estimate of 'typical' park conditions and activity. However, as is also evident from the graphs, 'typical' conditions are hard to find and define, and the forecast model falls short at capturing large outliers (e.g. festivals, popular events, etc), and further capturing micro-variation in SocM activity, from using weather and temporal data alone. This model limitation was expected: As no reliable real-time data source was found that captured planned events, this parameter was not included in the model, and therefore it is expected that any day with a special event will be mis-predicted by the model.

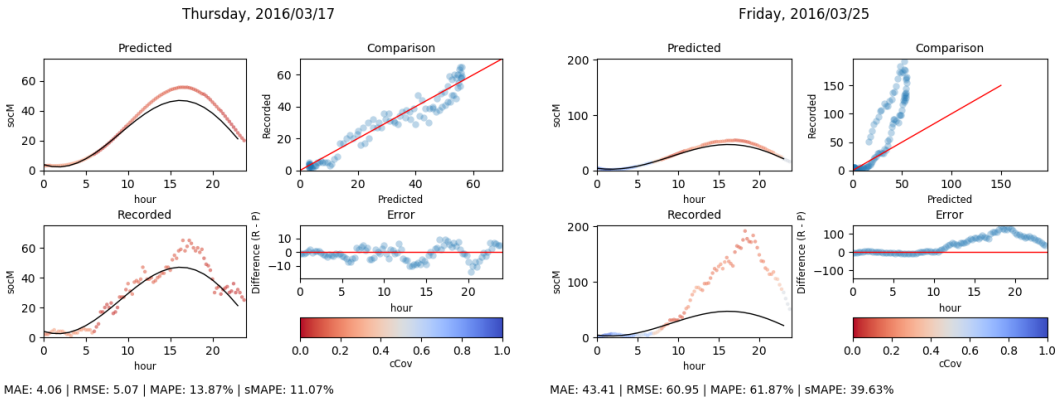
In validating the SDM, the goal was to evaluate the distribution of activity throughout the area only, and not focus on any temporal properties. Sub-model evaluation was performed against activity distribution in the area as captured in site surveys

Date	MAE	RMSE	MAPE	sMAPE
01/03/2016	7.750	9.234	39.06%	27.32%
02/03/2016	9.174	11.394	50.21%	24.62%
03/03/2016	4.248	5.547	15.26%	14.88%
04/03/2016	4.630	5.755	16.75%	18.89%
05/03/2016	10.769	14.791	45.03%	24.40%
07/03/2016	6.222	7.897	27.23%	18.79%
08/03/2016	6.707	8.721	29.71%	17.62%
10/03/2016	4.799	6.543	19.19%	16.71%
11/03/2016	8.018	12.899	22.36%	15.39%
12/03/2016	16.015	20.211	33.44%	23.55%
13/03/2016	16.917	25.034	25.99%	22.55%
14/03/2016	12.844	17.128	33.31%	29.51%
15/03/2016	7.560	9.023	29.30%	24.52%
16/03/2016	5.844	7.305	22.28%	18.99%
17/03/2016	4.058	5.074	13.87%	11.07%
18/03/2016	4.639	5.563	21.36%	21.03%
19/03/2016	6.195	8.025	25.02%	22.83%
20/03/2016	9.238	11.900	23.43%	18.57%
21/03/2016	6.690	7.949	25.34%	27.26%
22/03/2016	8.870	12.321	25.58%	16.76%
23/03/2016	7.033	8.883	26.93%	19.47%
24/03/2016	9.810	12.928	49.67%	24.28%
25/03/2016	43.408	60.947	61.87%	39.63%
26/03/2016	12.192	16.062	32.06%	27.06%
28/03/2016	9.231	13.226	28.66%	25.71%
29/03/2016	6.389	9.239	25.62%	19.58%
30/03/2016	7.349	9.186	24.24%	19.98%
31/03/2016	12.388	16.452	32.78%	23.31%

Table 8.7: Forecast Model Validation - HyP

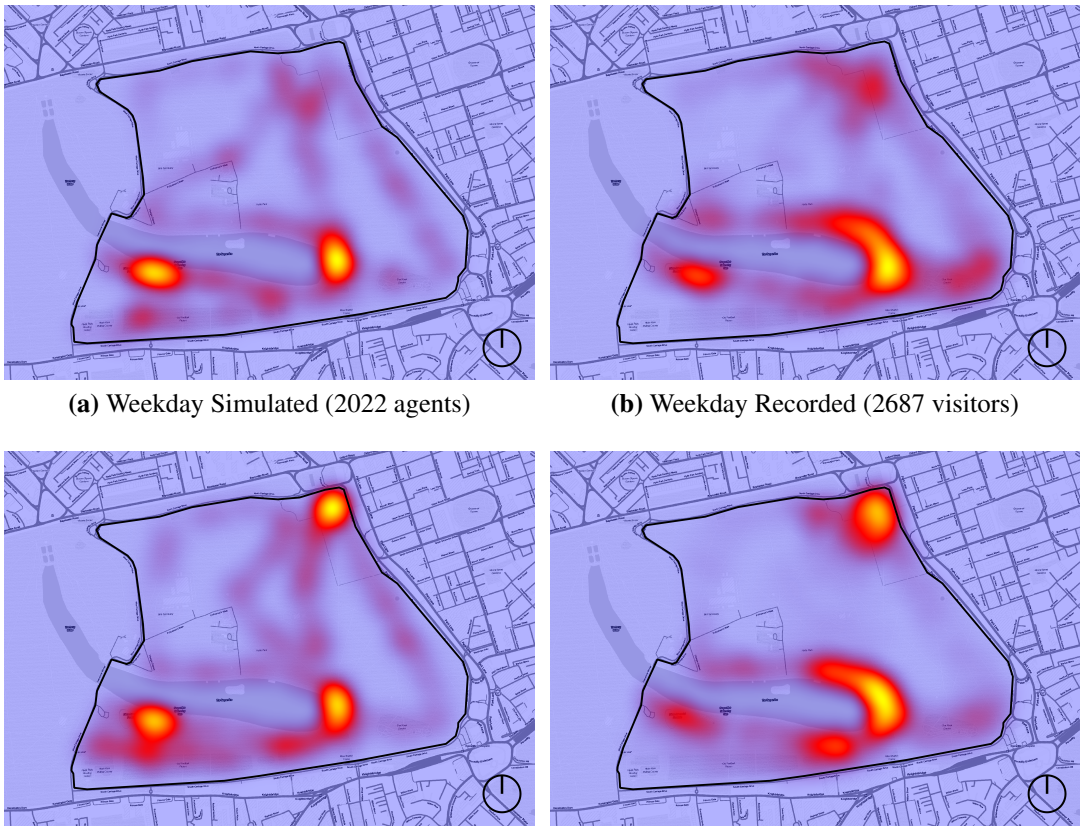
performed on 02/10/2015 and 11/10/2015. Synthetic data was collected by running a simulation with an agent population size similar to visitor volume collected at each survey. An initial comparison between simulated and recorded activity distribution is presented in Figure 8.25, visualising activity heatmaps, which shows that the model is successful in replicating major activity hotspots.

Further to visual evaluation, a statistical validation method was implemented, in order to apply a more thorough comparison. The method used was the *Expanding Cell Validation Method*, proposed by Malleson et al. (2010). The expanding cell vali-



(a) Most accurate 17/03/2016 **(b)** Least accurate 25/03/2016

Figure 8.24: Daily SocM Forecasts - Validation



(a) Weekday Simulated (2022 agents) **(b)** Weekday Recorded (2687 visitors)

(c) Sunday Simulated (4430 agents) **(d)** Sunday Recorded (4340 visitors)

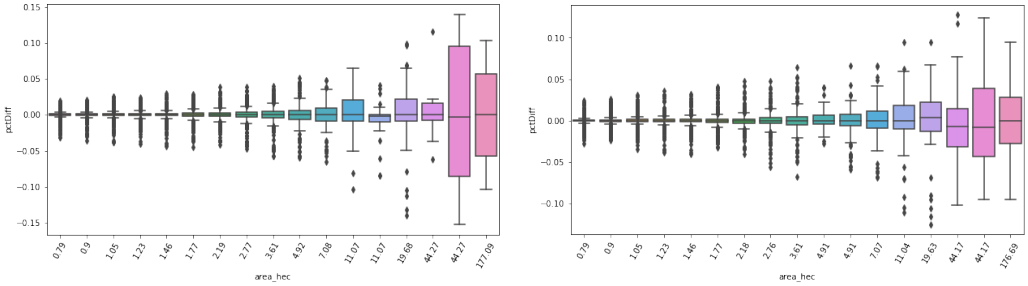
Figure 8.25: Hyde Park ABM Activity Heatmaps

dation method involves aggregating points to a regular grid duplicated and shifted slightly in the four cardinal directions, and measuring the error between simulation and observations for each cell. Additionally, given that simulation results may not

always replicate total recorded activity accurately, proportional values are used for each cell taken as a percentage of the total observation count. These two characteristics present two benefits of the method: Proportional counts allow for comparison between datasets of different size, and the shifting grid helps account for the modifiable areal unit problem, as each location is potentially measured by up to five different grids, therefore highlighting the validity of any hotspots.

For the Hyde Park case study, multiple grid cell sizes were generated and tested, to observe how the model behaves at different observation scales, and to identify the scale of accuracy for the model. The code used to generate the grids and calculate the cell error is presented in Appendix A.5.2. Results are shown in Figure 8.26. As can be seen, the model's effective scale of accuracy is found to be better at smaller observation scales (Figures 8.26a, 8.26b). This is to be expected; the majority of activities is found to be concentrated around points of interest both in the simulation and in observations, specifically restaurants and the Speaker's corner. Although the simulation has captured the *locations* of these hotspots accurately, the *volume* of activity was not captured in the simulation[2], with the model either over- or underestimating the magnitude of activity in these locations. At larger grid sizes, these hotspots are the major influence in any cell count capturing them, whereas at smaller grid sizes the grid capturing method is essentially overtaken by site survey recording accuracy, which was set to approximately 100 meters, and therefore at this scale point locations have been sufficiently randomly distributed to provide a smoothing effect over the cell. Given the scale of observation then, the ABM's effective scale of accuracy for CS1:HyP was identified to be at a cell size of 2.77 ha (the grid size closest to the observation size of 3.14 ha, as defined by the 100m radius observation range). At this scale, the ABM performed well in identifying activity hotspots as well as identifying areas with minimal activity, with overall accuracy error below 5.5%.

[2]And it was not expected to be captured either, as no detailed dataset was available regarding restaurant visitor numbers



(a) Weekday Simulation: Cell Error by Cell Size **(b)** Sunday Simulation: Cell Error by Cell Size

(c) Weekday Simulation Validation Grid - Cell Size: 2.77ha **(d)** Sunday Simulation Validation Grid - Cell Size: 2.76ha

Figure 8.26: Hyde Park ABM Spatial Validation. Error is measured as the difference between proportional grid counts ($recorded - simulated$). Red hues show model overestimation, blue hues show model underestimation.

8.6 Summary

Overall, CS1:HyP offered a valuable test-bed for applying and testing the various methodologies discussed in this work. Furthermore, it was carried out using solely publicly available datasets, and the interpretation of any results should be done with this constraint in mind, as this case study has demonstrated that physical public life is indeed reflected to a degree in our digital public life.

With specific regard to the aims set out at the beginning of this chapter:

1. **Identification of relevant data sources.** A range of publicly available, potentially real-time data sources that may reflect public space activity were identified and tested, including social media platforms *Twitter*, *Instagram*, and *Facebook*, weather forecast web service *forecast.io*, and transport data

from *TfL*. Of these, only weather forecast data was reliably captured throughout the duration of the study. Twitter and Instagram data was being reliably collected until changes in one of the services made collection impossible, and therefore while those datasets were relevant, they are no longer available. The rest proved to be either unreliable in collection, or irrelevant to public space activity.

2. **Development of appropriate data capturing methodologies.** The automated collection scripts presented here provided a reliable way of continuously capturing data for a duration of approximately 9 months. Although data collection was taking place once a day, the methodology could certainly be applied to shorter time spans to have a real-time data collection.

3. **Development of a PSA forecast model, capable of performing in RT, and subsequent calibration of the model using available data sources.** The aggregate park activity forecast model discussed here performed well in capturing 'standard' activity throughout the day for uneventful days, but was found lacking in capturing special events. Nevertheless, this forecast model was useful at continuously providing baseline estimations at 15 minute intervals.

4. **Development of a SDM using the ABM paradigm, to simulate individual visitor activity in the area of interest, capable of performing in RT. Subsequent calibration of SDM parameters.** The ABM used to simulate individual visitor activity performed well in capturing both locations of increased activity, as well as areas of reduced activity. Initial aims involved the use of additional spatially fine datasets for validation, however as no such dataset was found, validation was partially performed against site survey data. Nevertheless, the Hyde Park case study, demonstrated that ABM modelled after observations of public space users' behaviour was successful in capturing overall park activity.

5. **Evaluation of the overall RT model, as well as sub-models.** The real-time

modelling methodologies presented in this work were not evaluated as an overall real-time model in this study, as no dataset was found that had both a spatial and temporal resolution high enough.

In conclusion, this case study offers partial validation of the modelling approach, as it illustrates that current activity under normal conditions can be predicted through environmental and temporal parameters. Even though the spatial disaggregation part of the model was not extensively tested, this is acceptable, as this case study was used as an initial test-bed for the development of methodologies. As such, even if overall real-time modelling methodologies were left without validation, this case study offered valuable insight for the development of the ABM that was used for the whole of this work.

Chapter 9

Case Study 2 - Queen Elizabeth Olympic Park

This chapter will discuss the second case study carried out in this thesis, Case Study 2: Queen Elizabeth Olympic Park (CS2:QEOP), with the aim of developing a real-time model of Public Space Activity (PSA) for Queen Elizabeth Olympic Park (QEOP) in London, UK. It was carried out as a direct continuation and for validation purposes of Case Study 1: Hyde Park (CS1:HyP), by extending the methodologies developed in CS1:HyP and applying them to a new target area. The new area of interest provided some additional opportunities as well in the form of new datasets not available for CS1:HyP, specifically detailed records of wireless device connectivity to QEOP's wireless network, which provided a significant level of detail for real-time park activity. CS2:QEOP therefore also focussed on exploring the potential of new datasets from networked infrastructure for the purposes of real-time PSA models. This chapter will follow a similar format to chapter 8, and will cover the main aims and objectives of CS2:QEOP, an extensive discussion on datasets used along with collection methods and analysis, calibration of both model components (forecast model and Spatial Disaggregation Model (SDM)) in the context of the new area, and an evaluation of the overall model as well as its individual parts.

The chapter will begin with an introduction to the case study (*section 9.1*), establish-

ing the overall aims of CS2:QEOP. The area itself is presented initially, discussing particular characteristics regarding park use, management, and unique features and management goals, as well as area morphology and its effect in this case study. Following the area introduction, case study aims are presented to establish the research context, and the individual objectives for CS2:QEOP are set, which will guide the rest of this case study.

The next section (*section 9.2*) focusses on data used in this study, including both remotely captured Real-Time Data (RTD) and static/ground truth data. It re-introduces all data sources previously used in CS1:HyP (Social Media (SocM), weather, site surveys) and re-establishes them within the context of CS2:QEOP, and further introduces and discusses new datasets used in this case study (wireless connection records). This section also presents the data clean-up process and concludes with the final data formats used for the rest of this case study.

Following the presentation of datasets, the next section (*section 9.3*) discusses how those datasets were used to calibrate the various aggregate activity forecast models used to estimate current overall activity in QEOP. Given the multiple data sources available for this study, multiple different forecast model parameters are tested, using previous methods (SocM), new datasets (Wifi), as well as a naive forecast model, establishing the effectiveness of each.

Next, *section 9.4* presents the SDM model used to simulate individual visitor activity in QEOP. The SDM used here is the PSA Agent-Based Model (ABM) first presented in chapter 7 and applied in CS1:HyP, calibrated to QEOP parameters.

Finally, the second-to-last section (*section 9.5*) presents an evaluation for the whole case study. It discusses the validation methods used to evaluate both the overall model as well as its individual components (forecast model and SDM).

9.1 Overview and Aims

The second case study focusses on Queen Elizabeth Olympic Park (QEOP), a newly developed park in east London, UK. It was constructed for the London Olympics in 2012, and includes some of the most prominent buildings of the Olympic Games, such as the Olympic Stadium, the Aquatics Centre, and the cycling track (VeloPark). These buildings are surrounded by grasslands, meadows, and other open spaces comprising the overall park in an area spanning approximately 100 hectares, through which runs a significant waterway, the River Lea. The entity responsible for maintaining, developing, and delivering the park to the public is the London Legacy Development Corporation (LLDC).

The LLDC is aiming at developing a *Smart Park* at QEOP, by targeting contemporary challenges such as crowd management, environmental sensing, sensing local biodiversity health, and engaging visitors with the park, by exploring the potential applications of the Internet of Things (IoT) to address these challenges. These efforts are done in collaboration with academic and research institutions, including University College London (UCL), Centre for Advanced Spatial Analysis (CASA), and Intel Collaborative Research Institute (ICRI). These institutions in collaboration with the LLDC are developing state-of-the art methodologies to tackle the specific challenges mentioned here using novel datasets generated through these technologies.

CS2:QEOP fits within the greater context of QEOP as a smart park, focusing on crowd and park visitor activity captured through visitor connectivity to wireless network access points, in addition to employing previous visitor modelling approaches. For the purposes of CS2:QEOP the area boundaries are set so that the park includes mainly open, accessible spaces, as can be seen in Figure 9.1. As has been discussed earlier in this thesis, this work focusses exclusively on open, public spaces, and for this reason it was decided that indoor areas such as the Stadium itself or the Aquatics Centre would not be included, as indoor spaces potentially require different methodologies in modelling crowd behaviour. Therefore, for the rest of this case study any

Figure 9.1: Queen Elizabeth Olympic Park Case Study Area Boundaries

reference to QEOP in general will refer to the open spaces (pathways, grasslands, lawns, meadows, bridges, paved areas, outdoor restaurant seating areas, etc.) found in the area, and are freely accessible and visible from other locations in the park.

The primary aim of this second case study was to further expand on the approaches and findings of the first case study, by extending the application scope on to a new location. Overall method validity has been established already, and its application to real-world scenarios has been examined in the previous case study and found to have acceptable results, for components where data was available for validation. Therefore, the specific objectives set for this second case study were different to the ones set for CS1:HyP. More specifically, CS2:QEOP was planned as a continuation and cross-validation to CS1:HyP with an additional focus on PSA information captured via novel sensing approaches implementing networked infrastructure, and furthermore as a test-case regarding ABMs of public space activity applied in morphologically more complex environments. The particular objectives were set as follows:

1. **Validation of CS1:HyP methodologies.** All of the successful approaches

developed and used in CS1:HyP were again applied in CS2:QEOP, to examine their applicability to a wider range of scenarios. Existing methodologies re-applied in this case study include the capturing of SocM and weather data, re-calibrating and running the real-time aggregate activity forecast model, re-calibrating the SDM to simulate park visitor activity in QEOP, and implementing similar validation methods regarding sub-model performance.

2. **Exploration of the potential of novel datasets.** QEOP, through its 'Smart Park' approach, presents an opportunity in capturing novel information of park visitor activity, in the form of connections to various access points of the wireless network at the park. These new, non-publicly available datasets offer a potentially much more varied and detailed picture of park visitor activity, and are furthermore captured in real-time, and therefore their potential contribution in Real-Time Simulations of Public Space Activity will be examined.

3. **Verification of PSA ABM capabilities in more complex environments.** This case study presents a more complex overall geometry compared to CS1:HyP, and therefore allows for the evaluation of ABM performance in more varied environments. Compared to the environment in CS1:HyP, CS2:QEOP has a steeper landscape with significant (for an urban park) hills and valleys, as well as overlapping geometry such as bridges. Overall model applicability to varied morphologies of urban space was evaluated by applying the SDM to a detailed virtual 3D reconstruction of QEOP.

9.2 Data Sources and Analysis

The entirety of the datasets used in the second case study will be discussed in this section, covering capturing methods, manipulation/clean-up, and uses. Overall, the datasets used are divided into two categories: Static data, containing data that was captured intermittently, data referring to a single point in time, or used as such, and Continuous/Real-Time data, containing datasets that were made available in

a streaming, real-time fashion, or data that can be potentially used in a real-time fashion, as defined in section 4.1.

9.2.1 Real-Time Datasets

9.2.1.1 Social Media - Weather Data

Social Media (SocM) data was collected for CS2:QEOP to be used as a proxy for real-time park visitor activity in the park, and correlated with weather conditions to investigate relationships between weather and park activity. Collection and analysis methodologies follow directly from the methodologies developed and applied in CS1:HyP. More specifically, SocM collection focused on geolocated *Twitter* and *Instagram* posts originating from within the boundaries of QEOP, and was performed using the respective services' Application Programming Interfaces (APIs). Collection began on January 28th 2016 and ended on May 31st 2016, after significant changes in Instagram's API, as discussed earlier in section 6.1. Weather data was collected using *darksky.net*'s weather API for the same period. The automated scripts used for SocM and weather data collection are presented in detail in section A.3 and section A.4 respectively. Out of the 125 days in total, on 1 day the collection script failed to run. Furthermore, a subset of the full dataset was removed and stored for validation purposes, spanning 30 days during the month of March 2016, 24.19% of the total dataset.

Initial results from SocM data collection around QEOP seemed to be consistent with similar data from CS1:HyP, with an average daily total of 323.9 SocM posts and value peaks generally centered around Sundays and Weekends, as expected. However, after initial clean-up and filtering of results by location, the final SocM dataset containing only records that originated from *within* the QEOP boundaries (rather than within a certain distance from the queried location, as is the standard response from platform APIs) was reduced significantly, with an average daily total of 76.3 SocM posts. A comparison of filtered and unfiltered results can be seen in Figure 9.2, where it is clear that valid results are a small fraction of the total dataset.

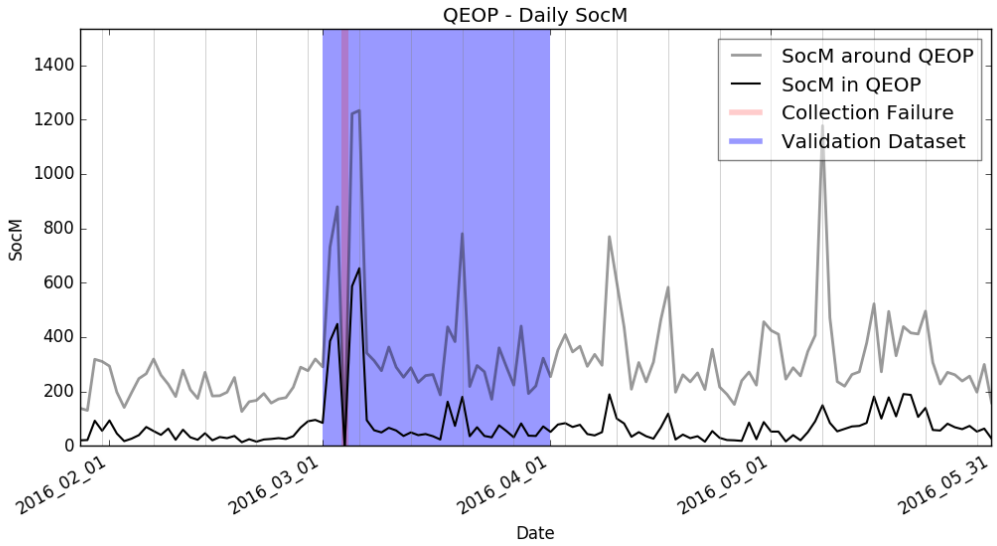


Figure 9.2: SocM Daily Totals In and Around QEOP

This difference between filtered and unfiltered values is attributed to two factors: First, the case study area's boundary geometry, which presents an elongated shape in the North-South axis, in combination with the respective platforms' search APIs which require a center point and search radius, result in a large circular catchment area centered on the park in order to capture all of the area, and therefore results for large parts outside the park are returned as well. Secondly, QEOP's close proximity to locations that consistently attract large crowds (such as the Westfield shopping centre and Stratford International train station) means that returned results include large crowds in close proximity to the park, but which do not necessarily translate into actual park visitors. Figure 9.3 illustrates these points, showing a map of the spatial distribution of filtered and unfiltered SocM events around QEOP for a single day.

SocM events were disaggregated to quarter-hour totals, and further smoothed using a moving sum window of 90 minutes, similar to CS1:HyP (Figure 8.6). As can be seen in Figure 9.4, quarter-hour values are quite low for the majority of days, with many days exhibiting zero SocM events even during midday. There does not appear to be any apparent pattern in variation throughout the day in SocM within QEOP, such as the one evident in CS1:HyP (Figure 6.11), and therefore this suggests that

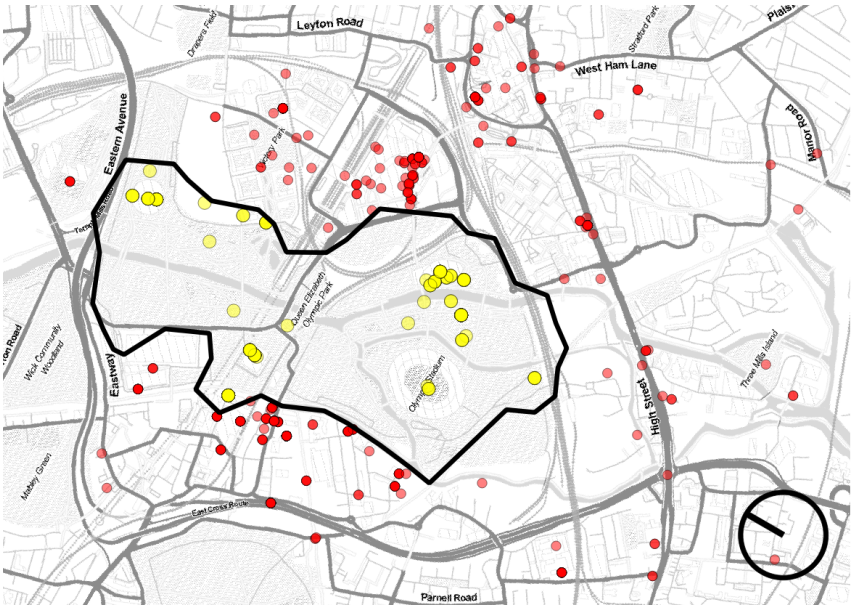


Figure 9.3: Spatial Distribution of SocM results in QEOP. All posts for a single day (2016/05/08) are shown, with points in yellow highlighting SocM posts in the area. Out of 1180 total records, only 164 are valid results.

quarter-hour SocM totals in this case study may not correlate with weather conditions for real-time predictions.

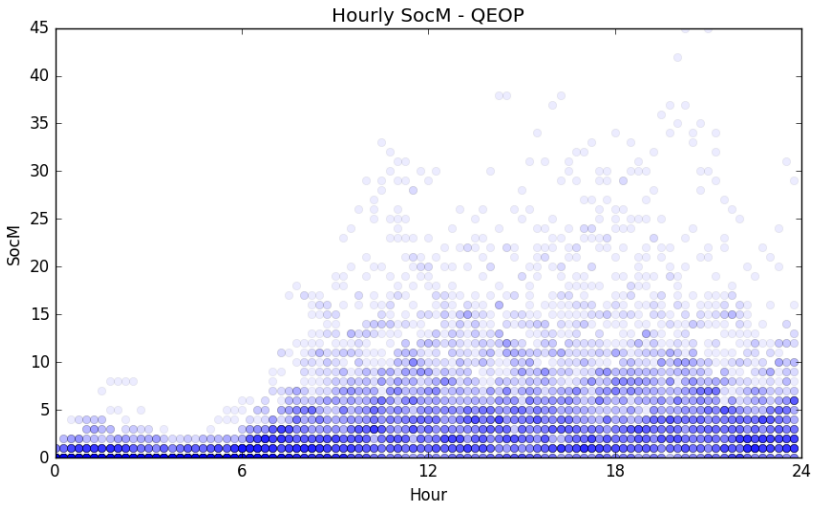


Figure 9.4: Quarter-Hour SocM Totals - QEOP (Individual points are plotted with opacity, more solid colors signify more occurrences)

9.2.1.2 Wireless Connections Data

In addition to the publicly available SocM dataset discussed above, an exclusive dataset pertaining to visitor activity was also examined in CS2:QEOP, which captured device connections to QEOP's wireless network. This WiFi dataset presented a detailed picture of real-time activity in the park, containing both spatial and temporal information on park visitor activity (i.e. where were individual visitors, as well as when they were there). For this case study, a sample of the dataset was used, which contained all records or the month of March 2016.

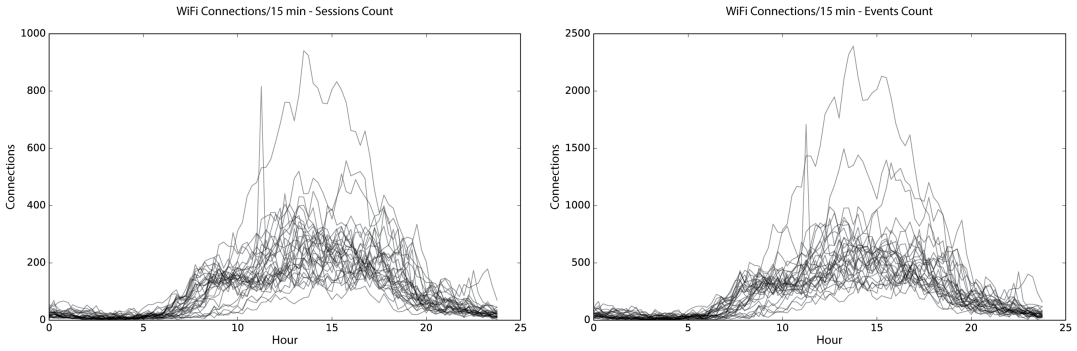
The dataset was formatted as a list of all individual connections, presented both as full connection sessions as well as connection event types, the difference between the two being that a session refers to the full connection for a single device from the moment it joined the network to the time it was disconnected from it (and thus a single record for the whole duration), while events captured the different interactions between device and network, such as 'JOIN', 'DISCONNECT', 'ROAM FROM', and 'ROAM TO'. Therefore a session typically has a unique id and represents the overall visit, and is itself comprised of multiple events. Additional information recorded in the dataset includes the unique session id, session end date and time, duration, data volume sent and received, and the access point id at which the device was connected during each event. A supplementary dataset included the access point ids and locations in the park.

Using the information offered in the WiFi dataset, 3 potential benefits were identified: First, it is possible to count the number of currently active connections for any point in time, and also to further deduce any derivatives, e.g. daily totals. Second, it is possible to count the number of currently active connections for any point in time *at each individual access point*, in other words it is possible to spatially disaggregate the dataset. Third, it is possible to track individual devices' movement throughout the park, recorded as transfers between access points.

Regarding the manipulation and clean-up of the dataset, the first step was to remove

sessions with overall duration larger than 14400 seconds (4 hours). The 4 hour threshold has been identified in previous studies (Ipsos Mori, 2015b) as the maximum visit duration. Furthermore, by examining the dataset, some obvious outliers were identified with extremely long durations (e.g. 2422285.82 seconds, approximately 672 hours) which could not have been park visitors, rather it is hypothesized that they may represent stationary wireless devices. Filtering for sessions shorter than 14400 seconds resulted in 306,275 records, accounting for 98.69% of the full dataset.

Temporal disaggregation to quarter-hour totals was performed using both the *sessions* and *events* datasets. The *sessions* dataset contains end times and durations for each connection, and for each session a count was added to the quarter-hour the session ended, as well as each previous quarter-hour for its duration, for sessions with a duration larger than 15 minutes. For the *events* dataset, a count was added to each quarter-hour where an event was recorded that was not a 'ROAM FROM' event, on the assumption that if an event has been recorded, then it must have been fired from a device within the area, and therefore the individual is in the area at the time of the event. Counts for each day are shown in Figure 9.5.



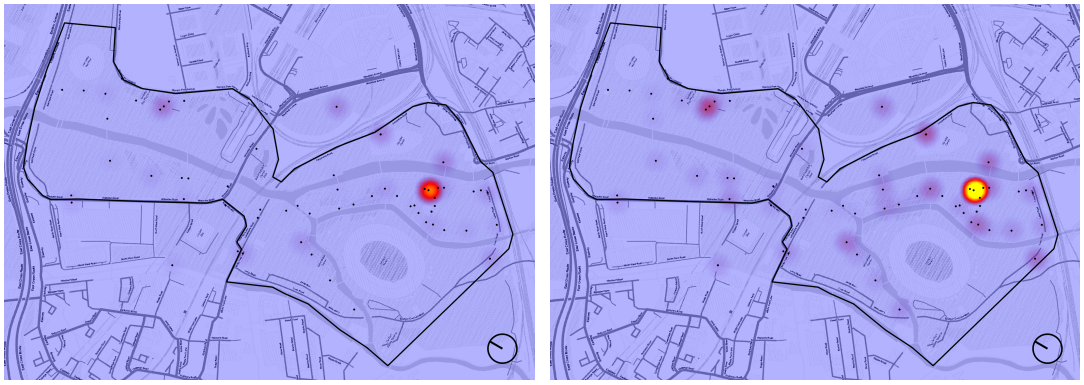
(a) WiFi Counts - Sessions **(b)** WiFi Counts - Events

Figure 9.5: WiFi Counts - Quarter-Hour Totals

Both approaches show similar curves for individual days, capturing matching peaks, however the *events* counts are on average 2.5 times larger that the *sessions* counts. This is expected to an extent: Each session is counted once for each quarter-hour of

its total duration, however the overwhelming majority of sessions are shorter that 15 minutes. On the other hand, multiple events may be contained within a single session (a typical session might include for example the initial 'JOIN' event, a few pairs of 'ROAM TO' and 'FROM', and the final 'DISCONNECT' event), therefore inflating the count.

In a similar approach to temporal disaggregation, WiFi data was disaggregated spatially as well, to capture the spread of activity throughout the area. For the spatial disaggregation, network access point ids were used to append connections and events to individual access points, whose locations are known. For the *sessions* dataset, all counts were appended to the access point the device disconnected from, as this was the only available data. For the *events* dataset, each event was appended to the access point that captured the event. Results are shown in Figure 9.6 for a sample day and time (2016/03/18, 14:30). In both cases, one particular access point (AP62) is capturing the majority of connections, 20% of all connections on average.



(a) WiFi Counts - Sessions **(b)** WiFi Counts - Events

Figure 9.6: WiFi Counts At Access Points (2016/03/18 - 14:30). Heatmaps with a Search Radius of 100, max value of 100.

The WiFi dataset schema has the potential to map individual devices' movement through the park, by tracking their hops between access points in the *events* dataset. However, examination of actual data illustrated a lack of consistency for this task: For example, the total count of 'JOIN' events in the dataset is 330891 events, while only 197820 'DISCONNECT' events are recorded, meaning that 133071 sessions

are not finalized. Additionally, multiple instances were identified were the same device id (essentially a unique device) fired multiple consecutive 'JOIN' events, with no 'DISCONNECT' events. For this reason, along with the observation from the spatial disaggregation regarding the accumulation of the majority of events to a single access point, it was decided that mapping individual devices' movement through the park was infeasible through the WiFi dataset.

9.2.2 Static Datasets

9.2.2.1 Site Surveys

Data on actual visitor activity in QEOP was collected on four occasions via site surveys, with the aim to record the locations and activities of visitors in the park as *'ground truth'* information. The method used for recording visitor activity is a slightly revised version of the site survey method used in CS1:HyP, and has been presented in detail in a previous chapter (*Section 6.3*).

Similar to CS1:HyP, an optimal path was planned throughout the area, that would cover the largest area at the shortest time, and any park visitors visible within approximately 150 meters from the surveyor path were captured. In contrast to CS1:HyP however, and due to the more complex terrain in QEOP compared to Hyde Park (HyP), the area was first divided in separate areas of varying shapes and sizes, indicated mainly by which locations where visible from the surveyor path. The path and survey areas can be seen in Figure 9.7.

Four surveys were carried out, on two consecutive days in August 2016. The surveys were carried out in afternoons, at what was assumed to be peak activity hours for QEOP. A summary of observations for each site survey is presented in Table 9.1. It appears that activity in QEOP is found to be lower than activity observed in CS1:HyP, with activity peaking in the mid to late afternoon (approximately at 2pm). A more detailed view of individual recorded activities (Table 9.2) provides an image similar to observed activity in CS1:HyP, with stationary activities accounting

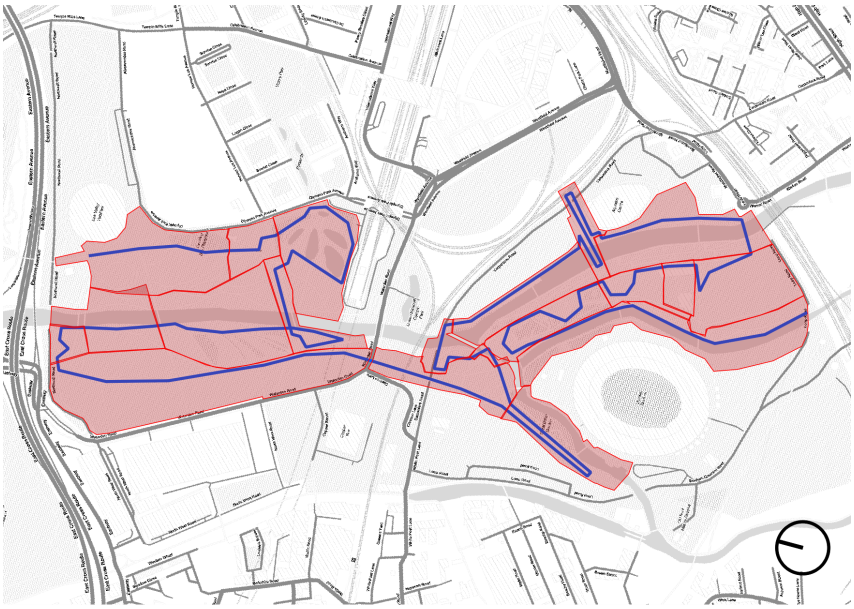


Figure 9.7: QEOP Site Survey Path and Areas. The dark blue line signifies the survey path, while the red polygons mark the different survey areas.

for slightly more than half of all observed activity (50-58% of total activity).

Site Survey	Date	Day	Start Time	End Time	Duration (minutes)	Total Visitor Count
QEOP-S1	2016/08/22	Monday	13:37:29	14:59:17	81	1520
QEOP-S2	2016/08/22	Monday	15:53:36	17:08:39	75	1479
QEOP-S3	2016/08/23	Tuesday	12:01:07	13:16:28	75	2008
QEOP-S4	2016/08/23	Tuesday	13:59:27	15:18:24	78	2656

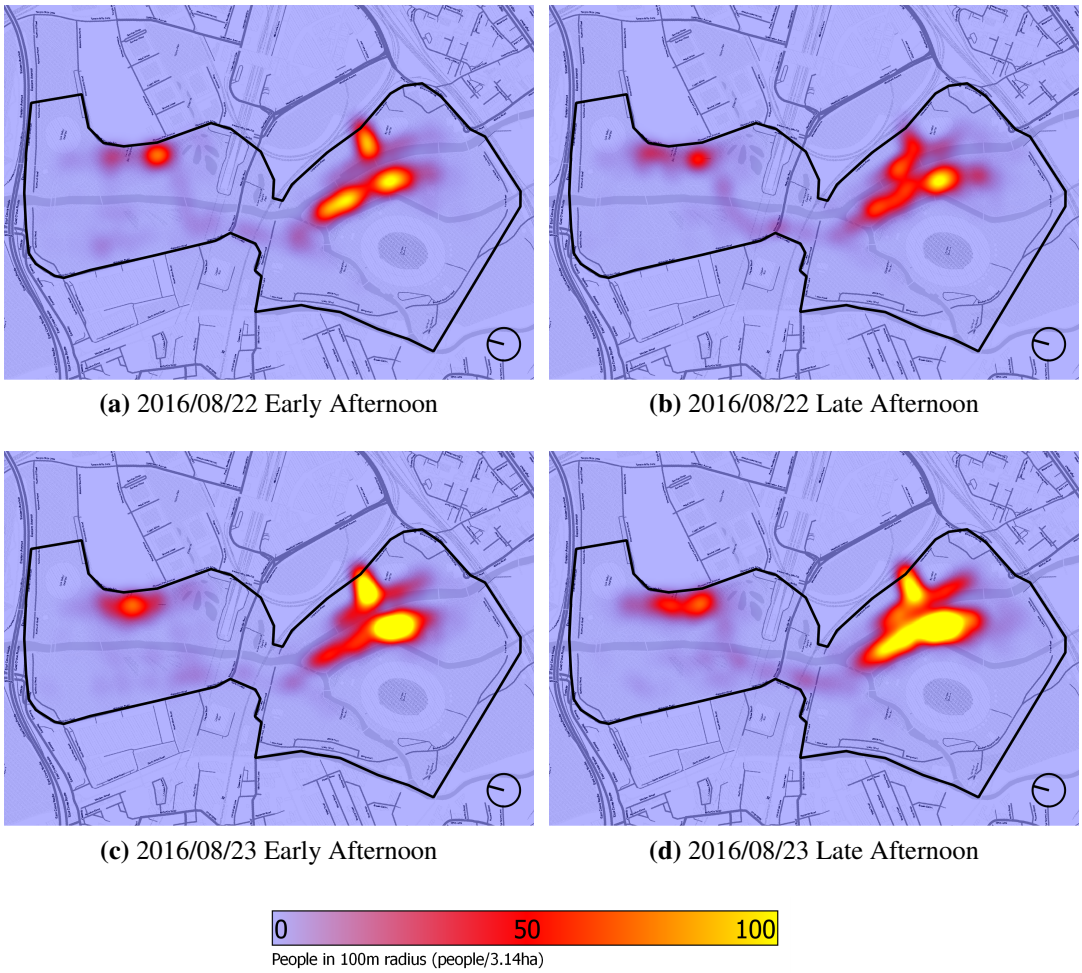
Table 9.1: QEOP Site Survey Summary

Site Survey	Date	Day	Total Visitor Count	Sitting Counted	Walking Counted	Walking Estimated	Sitting Percentage
QEOP-S1	2016/08/22	Monday	1520	1026	494	988	50.94%
QEOP-S2	2016/08/22	Monday	1479	1001	478	956	51.15%
QEOP-S3	2016/08/23	Tuesday	2008	1430	578	1,156	55.30%
QEOP-S4	2016/08/23	Tuesday	2656	1952	704	1,408	58.10%

Table 9.2: QEOP Site Survey Visitor Statistics

Further work on the filed survey data required the dispersion of recorded activity points from the surveying path into the general area. For the dispersion, the previously used distance limit was used, with a dispersion distance limit set to 150 meters. An additional rule was used however, relating to the survey areas mentioned earlier, so that the newly calculated activity location was also within the current survey area, adding another level of consistency between actual and recal-

culated activity locations. The final recalculated activity heatmaps can be seen in Figure 9.8



(a) 2016/08/22 Early Afternoon

(b) 2016/08/22 Late Afternoon

(c) 2016/08/23 Early Afternoon

(d) 2016/08/23 Late Afternoon

Figure 9.8: QEOP Site Survey Activity Heatmaps.

9.2.2.2 Webcam Pedestrian Counts

Another exclusive dataset that was used in CS2:QEOP is pedestrian counts performed at park entrances, captured using camera tracking and automated counts. Although in principle this dataset may be used in real-time, with current pedestrian counts being streamed directly into the model, in this case it was used as static data, to identify the most popular gates overall. This information was subsequently used in the SDM (as discussed later in section 9.4), to calibrate the gate weights for the park visitor ABM.

id	name	entries	exits	total	average
10	Canal Park Towpath	2166	785	2951	1475.5
8	F10 Bridge	122648	94643	217291	108645.5
6	GreenwayPath	3761	753	4514	2257
2	Hackney Wick Bridge	4246	32679	36925	18462.5
5	HonourLeaAvenue	45692	59142	104834	52417
7	Iron Bridge LH	8981	9987	18968	9484
9	MonierBridge	10680	14275	24955	12477.5
3	Top of Waterden Road	3453	3632	7085	3542.5
4	Westfield Avenue	27978	30816	58794	29397
1	WhitePostLane	11081	10454	21535	10767.5

Table 9.3: Pedestrian Entries and Exits at Gates

Pedestrian counts were supplied as totals at short intervals for each location, divided into people exiting and entering the park, for the whole month of March 2016. These values were summed for each location, and a weight for each gate was calculated as an average between entries and exits. The values are shown in Table 9.3. It is interesting to note that some gates exhibit significant differences between entries and exits: *F10 Bridge (8)* shows an abundance of entries of approximately 28,000, while *Hackney Wick Bridge (2)* exhibits a deficit of a similar level, meaning that some gates tend to function mainly as entrances while others are used predominantly as exits. Furthermore, this characteristic suggests that the park is also used as traversal space, especially given the fact that the most popular entrance is F10 Bridge (8), which is one of the park entrances closest to a major train terminal. The spatial distribution of total counts by gate can be seen in Figure 9.9.

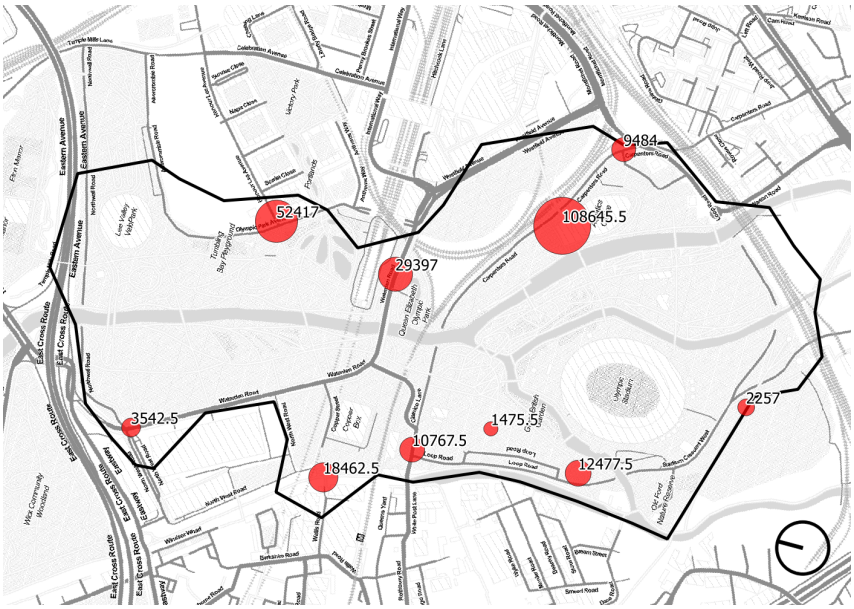


Figure 9.9: Pedestrian Counts at Gates - QEOP

9.3 Forecast Model

The real-time datasets presented in the previous section were primarily used in calibrating the aggregate activity forecast model for QEOP. It was assumed that these two datasets (SocM and WiFi) offer a representative sample of actual visitor activity, and as such could be used as a proxy for actual activity. Three forecast models were implemented: A Generalized Linear Model (GLM) correlating SocM as a function of weather and temporal conditions (same as the polynomial regression forecast model implemented in CS1:HyP), a similar GLM calculating WiFi connections as a function of weather and temporal conditions, and finally a naive forecasting model. All three models were implemented at a temporal resolution of 15 minutes.

9.3.1 Social Media - Weather Forecast Model

The SocM - weather forecast model that was implemented in CS2:QEOP is a multiple linear regression model, correlating quarter-hour SocM values (predicted variable) with time of day and weather parameters (predictor variables), as presented and implemented in CS1:HyP. Correlation with time of day was implemented as a polynomial regression, while weather parameters were included as scalars. In its

general form, the multiple linear regression model is defined as

$$SocM_t = b_0 + b_1 * hr_t + b_2 * hr_t^2 + ... + b_n * hr_t^n + W_t$$

for cases where weather and temporal variables are combined additively, or

$$SocM_t = (b_0 + b_1 * hr_t + b_2 * hr_t^2 + ... + b_n * hr_t^n) * W_t$$

for cases where weather and temporal variables are combined multiplicatively, with $SocM_t$ signifying SocM events at time t, hr_t is the time of day at time t, and W_t is the relevant weather variable at time t.

A series of calibration tests were performed, to identify the optimal parameter set and coefficients for the SocM/weather GLM, by using an Ordinary Least Squares (OLS) method and measuring the adjusted R^2 values for different model forms, parameter sets, and day type classifications. The calibration test results are presented in Table 9.4.

Model	**Week**	**Weekdays**	**Weekends**	**Saturdays**	**Sundays**
SocM : hr^3	0.192	0.164	0.288	0.293	0.282
SocM : hr^4	0.197	0.166	0.301	0.316	**0.288**
SocM : hr^5	**0.197**	**0.169**	**0.302**	**0.320**	0.287
SocM : hr^5 + temp	**0.235**	**0.216**	0.330	0.328	0.345
SocM : hr^5 + cCov	0.198	0.169	0.332	0.322	0.371
SocM : hr^5 + wndSpd	0.199	0.169	**0.332**	**0.327**	**0.355**
SocM : hr^5 + precP	0.197	0.170	0.302	0.322	0.287
SocM : hr^5 + precInt	0.198	0.171	0.302	0.319	0.287
SocM : hr^5 * temp	**0.247**	**0.238**	0.345	0.332	0.378
SocM : hr^5 * cCov	0.200	0.171	**0.361**	**0.366**	**0.421**
SocM : hr^5 * wndSpd	0.203	0.182	0.349	0.349	0.379
SocM : hr^5 * precP	0.201	0.173	0.313	0.357	0.292
SocM : hr^5 * precInt	0.200	0.174	0.310	0.348	0.288

Table 9.4: Adjusted R^2 for SocM - Weather Linear Model by Coefficient - QEOP. Model best fits for each calibration stage and day type are highlighted in bold.

Similar to CS1:HyP, a 5th degree polynomial for the time of day parameter provides

the best fit overall, compared to 4th and 3rd degree polynomials, although not for all day type classifications. regarding the combination of weather and time variables, it seems that multiplicative combinations provide a better fit, as expected, compared to additive combinations. However, no single weather parameter appears to provide the best fit across all day types in either variable combination type. Specifically, it appears that temperature, cloud coverage, and wind speed all provide some significant improvement in model fit over different day types. Finally, and most importantly, adjusted R^2 values are very low overall ($R^2 < 0.42$, 0.273 average), compared to equivalent values in CS1:HyP, which hints at issues with the dataset itself.

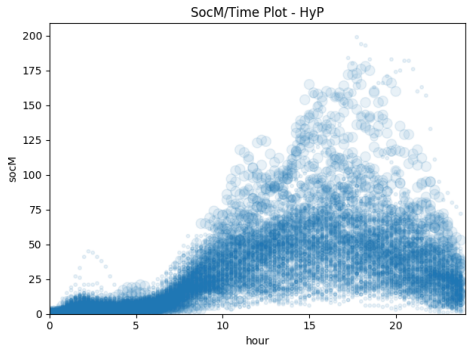


Figure 9.10: SocM/Time Plot - HyP

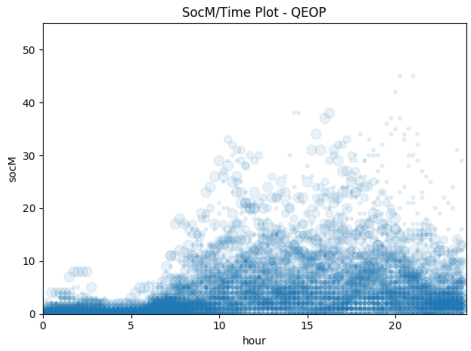


Figure 9.11: SocM/Time Plot - QEOP

Indeed, looking at a plot of SocM against time of day (Figure 9.11) and comparing to a similar plot for CS1:HyP (Figure 9.10), the reason for these low R^2 values becomes clear. First of all, SocM values for QEOP range between 0 and 50, whereas the HyP equivalent is 0 - 200. Secondly, although QEOP SocM follow a similar daily image to HyP SocM, with low values during the night and higher values found during daylight hours, they do not seem to do so with any discernible pattern, and in fact there appear to be zero values during the day as well. These two characteristics therefore appear to make forecasting SocM values using weather and temporal parameters in QEOP an ineffective approach.

9.3.2 WiFi - Weather Forecast Model

The second approach to forecasting aggregate park visitor volume in QEOP used the number of WiFi connections as a proxy for actual visitor activity. The dataset has been discussed previously (subsubsection 9.2.1.2), and it has been established that it is possible to extract total WiFi connections for any period of time. In order to implement a forecast model using WiFi data, the dataset was temporally aggregated to quarter-hour intervals, and subsequently a multiple linear regression model was developed, in a similar approach to the SocM/weather forecast model.

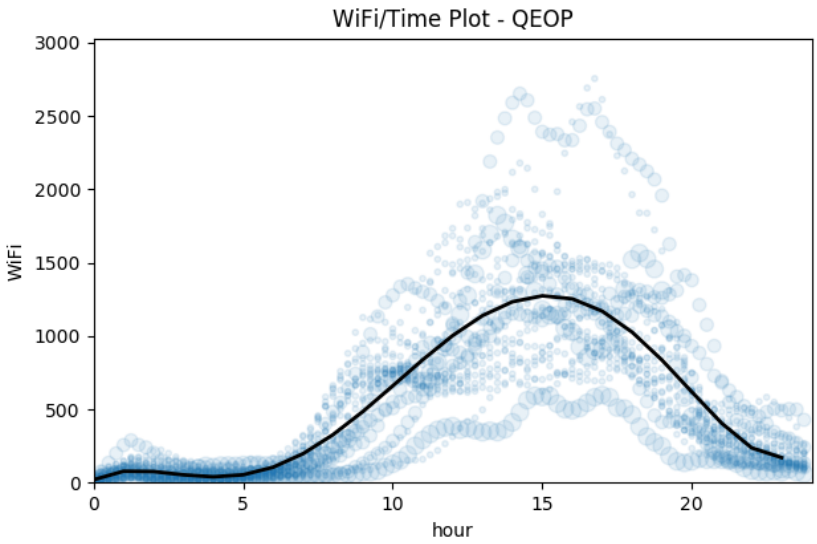


Figure 9.12: WiFi/Time Plot - QEOP, with trendline for 5th degree polynomial of time of day variable

Regarding the dataset itself, by plotting quarter-hour totals against time of day (Figure 9.12), it appears that the WiFi data exhibits a more distinctive and expected daily pattern compared to the QEOP SocM dataset, with values rising steadily from early morning into the afternoon, and then smoothly dropping into the evening, similar to HyP SocM values (Figure 9.10). However, the available dataset includes only a sample covering the entire month of March 2016, containing 31 days. Of these, 6 days were extracted and kept separately for validation purposes, thus reducing the total days to 25, which is evident in the thinned plot.

Model	Adjusted R^2
WiFi : hr^3	0.658
WiFi : hr^4	0.680
WiFi : hr^5	0.715
WiFi : hr^6	**0.716**
WiFi : hr^7	0.716
WiFi : hr^5 * temp	**0.731**
WiFi : hr^5 * cCov	0.718
WiFi : hr^5 * wndSpd	0.720
WiFi : hr^5 * precP	0.720
WiFi : hr^5 * precInt	0.717

Table 9.5: Adjusted R^2 for WiFi - Weather Linear Model by Coefficient. Model best fits for each calibration stage are highlighted in bold.

The multiple linear regression model implemented here was of the form

$$WiFi_t = (b_0 + b_1 * hr_t + b_2 * hr_t^2 + ... + b_n * hr_t^n) * W_t$$

similar to the SocM/weather forecast model. Given the sparsity of the dataset, it was not possible to classify by day type, as that would potentially leave the *Weekend* category with as little as 6 days' worth of data, and therefore the model calibration process was run for the whole week as a single set. The calibration process first looked at determining the polynomial degree with the best fit for the time of day variable, with the power of 6 giving the best adjusted R^2 result (0.716). For consistency, and given the fact that the 6th degree polynomial was marginally higher than the 5th degree polynomial (0.715), it was decided to continue the calibration process using the 5th degree polynomial for the time of day variable. Further comparison of adjusted R^2 values with multiplicative combinations of time of day and single weather parameters identified the temperature parameter as providing the best fit, with a value of 0.731.

9.3.3 Naive Forecast Model

In addition to the previous two aggregate activity forecast models which implemented existing and novel datasets to predict current activity, a naive forecasting model was also implemented. The naive model made use of the temporal nature of input datasets, providing predictions for each quarter-hour period based on the value of the previous quarter-hour period. Essentially, for each time step, the naive model assumed that the forecast value would stay the same as was it was when it was last reported. This approach was applied to both SocM and WiFi data. Due to its heuristic nature, no calibration was necessary for this model approach. The naive forecast model implementation for WiFi connections for a sample day is illustrated in Figure 9.13.

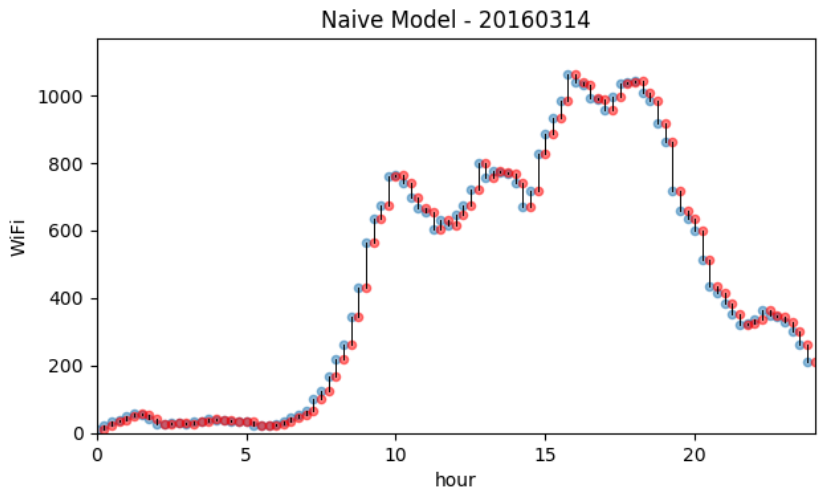


Figure 9.13: Naive Forecast Model for 2016/03/14. Blue points mark the observed values, red points mark the predicted values (equal to the previous timestep's observed value), black lines mark the error between observed and predicted value.

9.4 Spatial Disaggregation Model

The next step in the overall real-time park activity model involves the disaggregation of total activity into individual visitors, accurately dispersed in space. This was performed using the ABM of PSA presented in *chapter 7: Modelling Spatial Behaviour*, and is the same model applied in CS1:HyP. This section will discuss the ABM applied in QEOP. It will focus on two points: First, a discussion on the

generation of the 3D environment for the model will be presented, covering the process used here as well as potential alternatives and the reasons they were discarded. Second, the calibration process of the ABM itself in the context of QEOP will be discussed.

9.4.1 Virtual Environment

As has been discussed extensively in this work, it is important to include the third dimension as a core element of the models developed here. Due to the application scale and scope, which focus on crowd spatial behaviour at the human scale, environmental characteristics of volume and shape can have a significant effect on model behaviour, considering for example the effect that aspects such as slope and line of sight can have on human spatial interaction. With this in mind, it was decided that a virtual model of QEOP would need to adequately capture the spatial characteristics of the area, and therefore a 3D virtual model would need to be used.

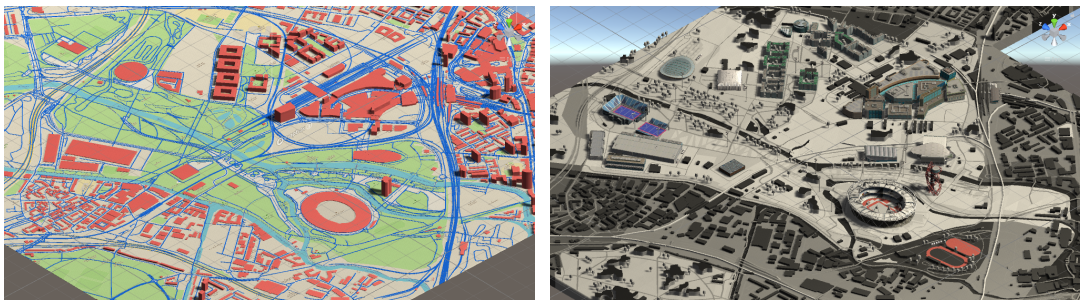
9.4.1.1 Procedural 3D Environment Generation

Initially, the use of existing 3D models was considered and potential alternatives were explored. In contrast to CS1:HyP where no existing 3D models of Hyde Park were available at the time of development, during the development of CS2:QEOP a potential source of 3D geometry was identified in procedural environment generation tools. These tools and platforms make use of the extensive Volunteered Geospatial Information (VGI) and web-mapping technologies currently available, and provide detailed 3D models of potentially any place on earth. Furthermore, various extensions and libraries exist which allow for quick implementation of 3D environment generation tools in multiple development platforms.

Three such services were identified that were of relevance to QEOP: *WRLD*[1], *Mapbox*[2], and *Mantle*[3]. All three provide tools for the procedural generation of 3D

[1] https://wrld3d.com/
[2] https://www.mapbox.com/
[3] https://www.mantle.tech/

environments, and furthermore all three are available as a package for the development platform used for the ABM here (*Unity*[4]). The Unity versions for Mapbox and WRLD were further examined as to their viability for generating the environment for QEOP. A bird's eye view of the resulting geometry from each package can be seen in Figure 9.14.



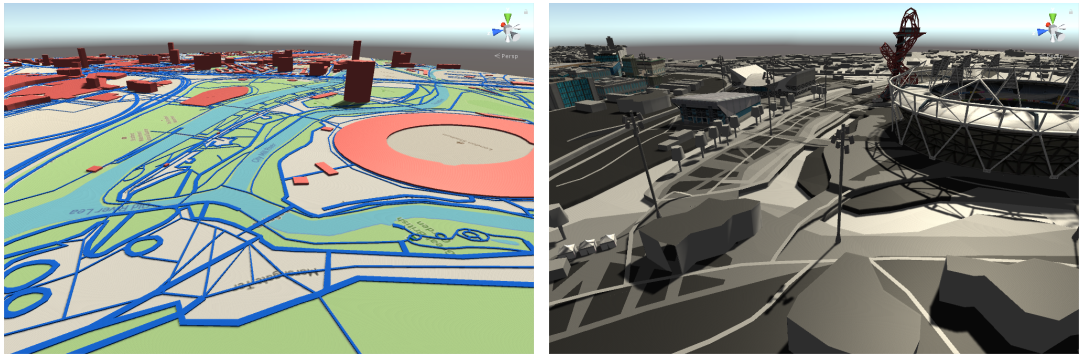
(a) QEOP in Mapbox **(b)** QEOP in WRLD

Figure 9.14: Procedural Generation of QEOP Environment.

On first inspection, both packages seem to provide good quality results, with continuous terrain geometry, and roads, buildings, and features as individual objects. One primary difference between the two services is regarding the underlying dataset each uses, as Mapbox makes use of open data (from OpenStreetMap (OSM)), whereas WRLD uses a set of proprietary and open data sources (for London these include the Ordnance Survey and OSM, along with data submitted by users/customers of the service). This is further evident in the richness of each result, where WRLD appears to have many more features, whereas further geometry features in Mapbox are available by extending the request (9.14a shows a result of the default request settings).

On closer inspection however (Figure 9.15), major issues were identified with each service. In WRLD's case, its data acquisition process appears to make it difficult (or at least slow) to update the underlying dataset: for QEOP, the generated environment appears to be a (highly accurate) representation of the state of the park during the Olympic Games in 2012 (5 years out of date at the time of writing), and furthermore its use of proprietary datasets makes it impossible for a user to change

[4]https://unity3d.com/



(a) QEOP in Mapbox - Detail **(b)** QEOP in WRLD - Detail

Figure 9.15: Procedural Generation of QEOP Environment - Detail.

an element directly (in contrast to using OSM data, which is open to editing for any registered user). In the case of Mapbox, although 3D geometry is generated with adequate detail in terms of features and recentness, it appears to be lacking in terrain fidelity, with the ground appearing to be almost flat, which in the case of QEOP is not, as the terrain presents an interesting relief, and is one of the main spatial interests of this case study. Therefore, for the reasons discussed here, it was decided that no existing procedural environment generation solution provided an adequate immediate result for the purposes of this study.

9.4.1.2 Virtual Environment Generation Process

Given that no procedural environment generation tool provided adequate results, it was decided that the QEOP virtual environment would be created manually. The model was created using data from using data from the UK Ordnance Survey and OSM, specifically the OS *Terrain 5 DTM* (Ordnance Survey Digimap Licence), and the OSM geodatabase (©OpenStreetMap contributors). The OS Digital Terrain Model (DTM) was used to generate the terrain elevation. The OSM geodatabase was used for a range of elements: line geometry for paths, polygon geometry for river boundaries, and point and polygon geometry for trees and wood areas. The 3D model was created using a series of software: QGIS, Autodesk 3DS Max, and finally Unity.

As a first step, the DTM raster files (9.16a) were imported into 3DS Max as grayscale heightmaps using a macro script[5]. Using image brightness values as height, a mesh was generated, and further optimized to reduce the number of polygons, using Delaunay Triangulation (9.16b). The resulting mesh (9.16c) was used as the base layer onto which the rest of the geometry was projected and created from.

The next step involved the creation of path and river geometry. This was achieved using a *'cookie cutter'* approach in 3DS Max: First, river bank geometry was extracted directly from the OSM geodatabase, and path geometry with path widths was similarly extracted using Mapbox. Second, the geometries were converted into closed polylines (polylines with no open ends, essentially a boundary line). Third, the polylines were projected onto the terrain mesh along the vertical axis, leaving an exact imprint. Finally, the interior mesh as defined by the imprint was cut and extracted as a separate object. An example of the closed polylines, mesh, and resulting path geometry can be seen in Figure 9.17.

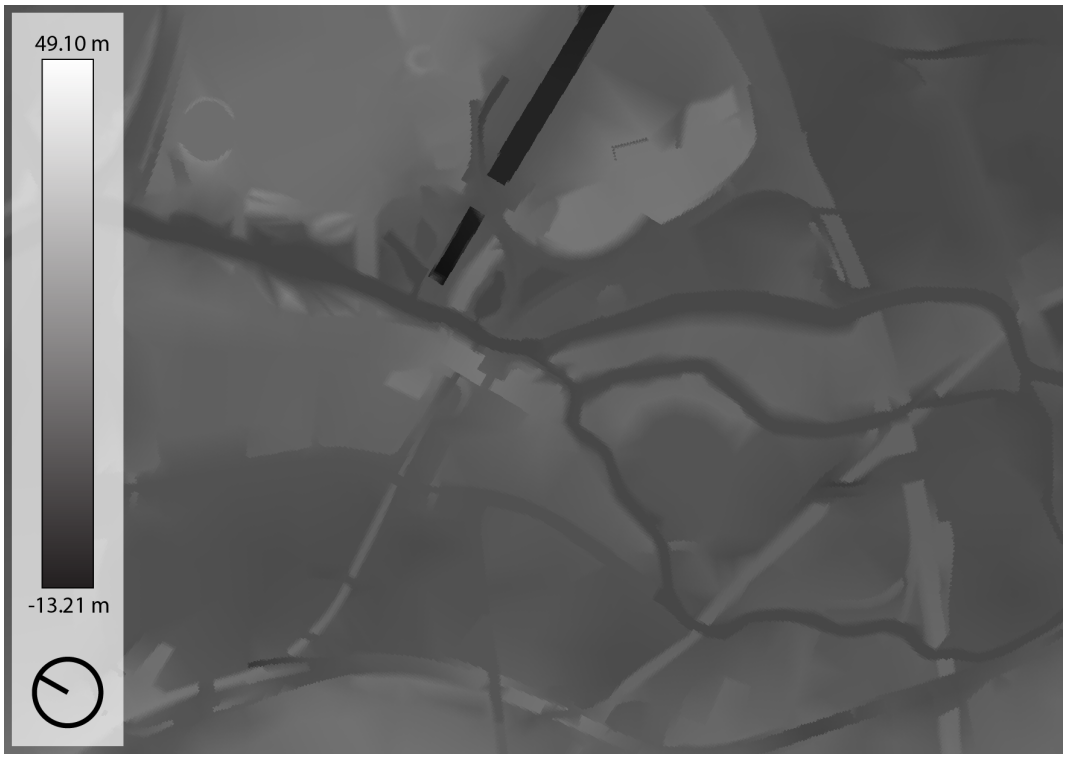
After creating the terrain and path geometries, the 3D model was imported into Unity for final adjustments and use. Bridges were manually placed in Unity to connect the land masses between rivers, gate locations were decided based on visits and official maps[6], and trees were positioned in the park. Tree locations were calculated using the same procedure as in CS1:HyP outlined in section A.5. The individual layers along with the final model can be seen in Figure 9.18.

9.4.2 Model Calibration

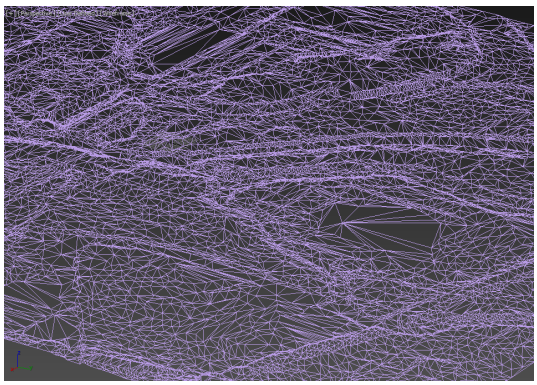
After establishing a working virtual environment, the implementation of the SDM and simulation of visitor activity becomes possible. This was performed via an ABM of park visitor activity. Model mechanics have been described at length, first

[5]script and directions found here: https://knowledge.autodesk.com/search-result/caas/sfdcarticles/sfdcarticles/Using-GeoTIFF-files-in-3ds-Max-and-Autodesk-VIZ.html (accessed 2017/09/01)

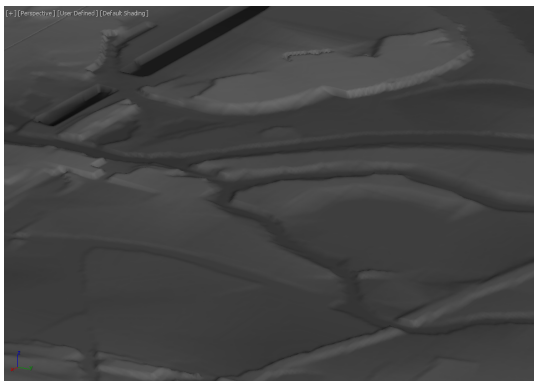
[6]http://www.queenelizabetholympicpark.co.uk/the-park/plan-your-visit/park-map (accessed 2017/09/01)



(a) QEOP DTM



(b) Terrain Mesh Wireframe (3DS Max)



(c) Terrain Mesh Surface (3DS Max)

Figure 9.16: Terrain Mesh Generation - QEOP

presented in chapter 7, and again applied in CS1:HyP (chapter 8). This section will focus on the ABM calibration process aimed at identifying the model parameters that best capture and recreate visitor activity in QEOP.

The calibration process was performed against visitor location data captured via site survey (Figure9.8b). Multiple values were tested for each parameter, by iteratively running the simulation while tweaking one parameter at each run, and

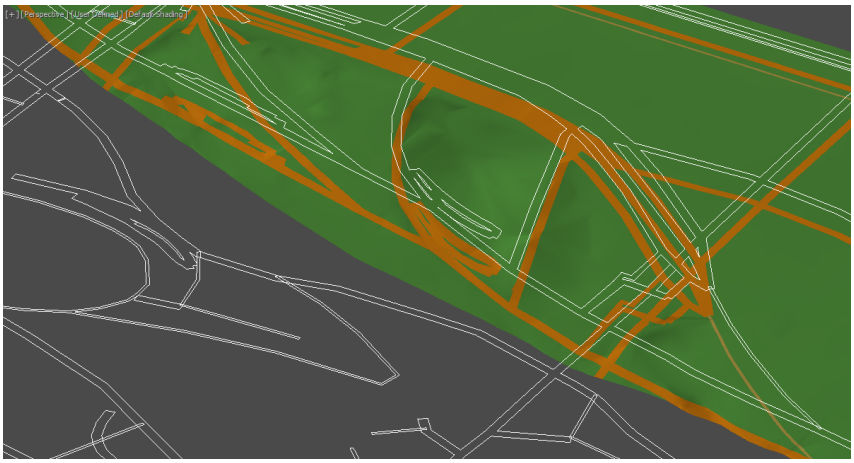


Figure 9.17: Path Geometry Creation - QEOP

recording model error at each iteration. Each calibration run was left to execute for 5000 frames with a fixed population of 1000 agents, corresponding to approximately 1700 park visitors, roughly similar to the 1480 visitors recorded on the day of the survey.

For the purposes of recording model error at each run, an error measure was developed which captured the average mean relative percentage error. It is a simplified version of the *Expanding Cell Validation Method* (Malleson et al., 2010), operating at a fixed grid size. For a set of locations y, the error between observed y_i and simulated $y_i^{'}$ values at each location i is measured as

$$\varepsilon_i = \left| \frac{y_i}{\sum y} - \frac{y_i^{'}}{\sum y^{'}} \right|$$

The mean of all relative percentage errors from all locations for a specific timestep t was considered to be the model Mean Relative Percentage Error (MRPE) for that timestep, so that

$$MRPE_t = \frac{\sum_i \varepsilon_i}{n_l}$$

where n_l is the number of locations. Model MRPE was captured at regular intervals (for a total of n_t recordings) during the calibration run, and a final average score of

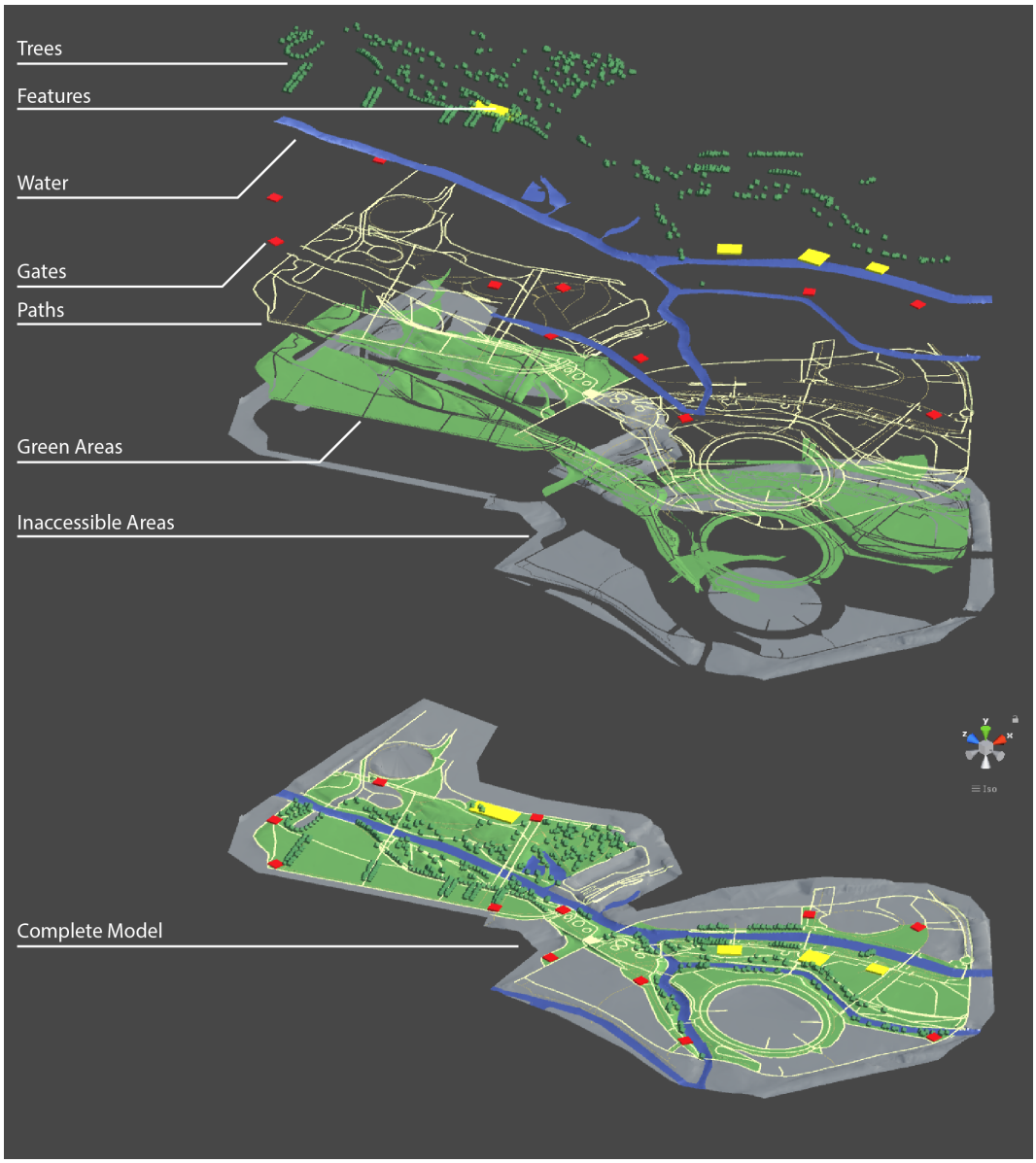


Figure 9.18: QEOP 3D Environment in Unity

all MRPEs of a particular run was considered to be the error measure for that run:

$$MRPE = \frac{\sum_t MRPE_t}{n_t}$$

For a given parameter set, the model was run twice, and the final score was calculated as the average of the two runs.

For the calculation of the error in the model, a square grid with cell size of 100m was generated so that it covered the entire case study area. The grid cell centroid

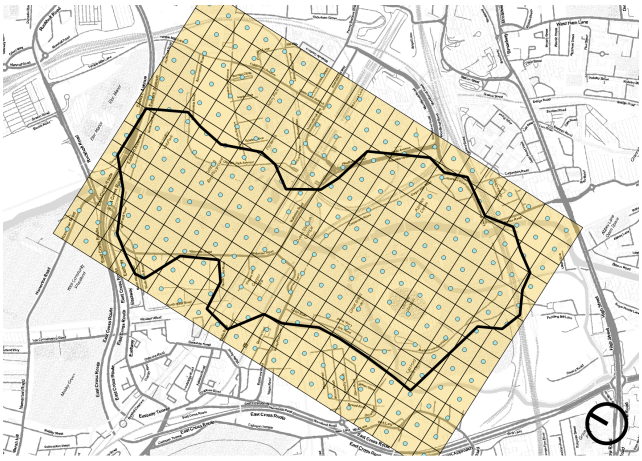


Figure 9.19: QEOP ABM Calibration Grid

coordinates were then extracted (Figure 9.19), and imported into Unity to be used as sampling locations. In the simulation environment in Unity, grid cell locations that did not overlap with the case study area were filtered out. Each of the remaining locations was set up to capture simulated activity and compare it to observed activity (Figure 9.20).

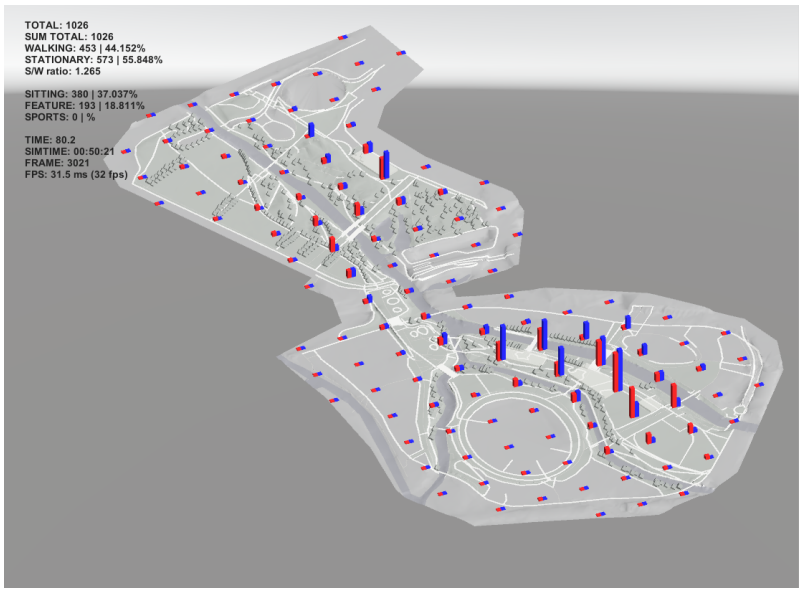


Figure 9.20: QEOP ABM Calibration Grid in Runtime. Blue colored bars highlight observed activity, red bars capture simulated activity.

During simulation runtime, current model performance was visually verified via a visualisation grid corresponding to the grid cell locations, set to visualise current error values. The points were set up to differentiate between model over- and under-

estimation using color. Error magnitude was communicated using color opacity and bar height, the former constrained to a range between 0%-10%, the latter unconstrained (Figure 9.21). Using this convention, where no bars are visible, the measured error value at that area is near zero, and therefore the model is accurately capturing activity at that location.

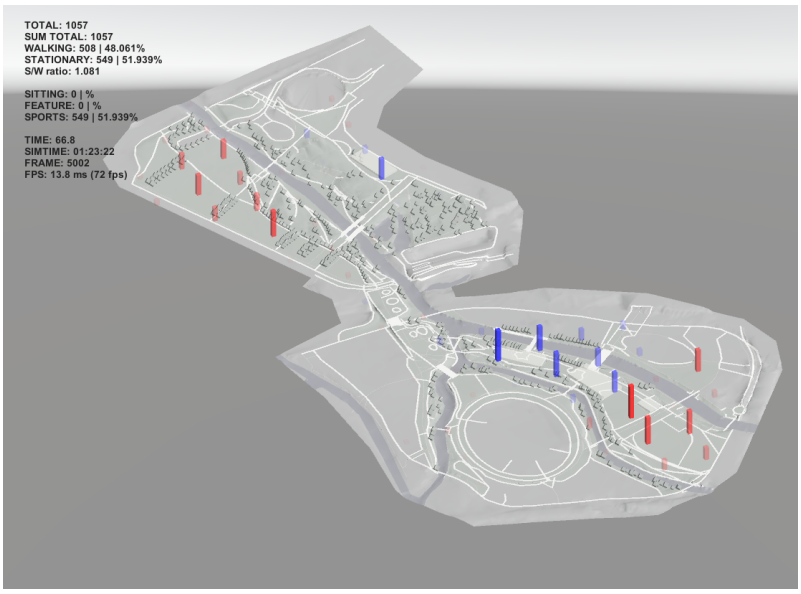


Figure 9.21: QEOP ABM Error Visualisation in Runtime. Red bars signify model overestimation, blue bars signify model underestimation. Bar height and opacity signify error magnitude.

The particular model parameters that required calibration in this case study were the gate weights, and the individual agent activity probabilities concerning the three main activities (Sit, Feature Visit, and Sports). For the first calibration run, in order to provide a baseline, no parameters were included, so that the model ran with all gates having equal probabilities to spawn an agent, and the agents only performed movement activities (Sit, Feature Visit, and Sports probabilities were set to zero). This produced an error score of 0.95%. The second run aimed to determine the effectiveness of gate weight inclusion, by adding attractors to each gate in the model derived from pedestrian traffic flow at each gate, as captured via CCTV. The specific values used are the ones shown in Table 9.3 and Figure 9.9. This resulted in a score of 0.89%, establishing the importance of gate preference for agents. The full list of calibration run scores can be seen in Table 9.6.

Model Parameters	Average MRPE
noParameters	0.95%
withGateWeights	0.89%
Sit60	0.88%
Feat60	0.50%
Sport60	**1.23%**
Sit20Feat20Sport20	0.70%
Sit30Feat30	0.51%
Sit30Sports30	1.05%
Feat30Sports30	0.76%
Sit20Feat30Sports10	0.56%
Sit30Feat20Sports10	0.63%
Sit20Feat40	**0.47%**
Sit40Feat20	0.55%
Sit15Feat40Sports05	**0.49%**

Table 9.6: QEOP Calibration Error Scores. Outliers are highlighted in bold, final parameter set is underlined.

The following calibration runs aimed to determine the particular probabilities for each agent activity type, and were all performed with gate weights enabled. From site surveys it was established that approximately 60% of recorded visitors were engaged in stationary activities, and therefore this was set to be the sum of all activity probabilities, so that approximately 60% of the agents in the model would be engaged in stationary activities at any point in the simulation. The initial set of model runs involved each individual activity being the sole activity for a particular run, and revealed that Feature Visits were the activity that provided the best fit, followed by Sitting and Sports activities in that order. This is not surprising, as during the site visits, the most crowded locations were the fixed-location restaurants, water fountain, and playground areas. Arguably, a majority of park users in QEOP visit the park for these specific locations, and can therefore be considered as fixed attractors in the area, set to attract agents to those particular locations, rather than letting

overall activity emerge procedurally. Through trial and error, it was determined that a mix of 20% sitting activities and 40% feature visits provided the best result. However a different parameter set was ultimately used (Sit 15%, Feature Visit 40%, Sports 5%). The reason for this choice is that during the site visits, sports activities were indeed spotted in all surveys (although not captured as such, only recorded as stationary/sitting activities), and therefore it was decided that sports activities should be included in the model.

9.5 Evaluation

At this point, all necessary components for the real-time simulation of park visitor activity in QEOP have been presented. Multiple forecast models have been developed that can provide continuous, short-term predictions of current and near-future overall activity in the park, using real-time Social Media (SocM), WiFi, and weather data as input. Additionally, a Spatial Disaggregation Model (SDM) of park activity has been developed and calibrated to capture individual visitor activity in QEOP, by implementing the ABM discussed in chapter 7. For the overall real-time model of visitor activity in QEOP, these two sub-models were combined so that forecasts were fed into the SDM, which then disaggregated those values into individual visitor activity in the park throughout the day.

In the following section the evaluation of all real-time modelling methodologies applied in CS2:QEOP will be presented. Evaluation was performed at two levels: first at the sub-model level, and subsequently at the overall level. At the sub-model level, the forecast model and the SDM were independently validated to ensure that they were working as expected. The evaluation process for both sub-models was similar to the evaluation applied in CS1:HyP. At the overall level, the full *Agent-Based Model of Real-Time, Public Space Activity* was evaluated in its entirety. This evaluation process was performed against novel datasets (specifically WiFi connectivity data) which was not available in CS1:HyP.

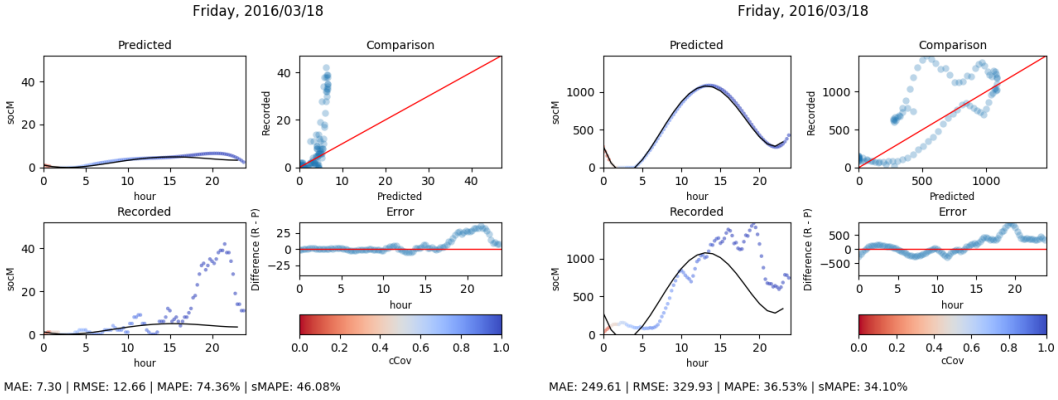
9.5.1 Forecast Model Evaluation

The forecast models developed in this case study and discussed in section 9.3 included two approaches, a linear model using weather conditions as predictors and a naive model, incorporating two different datasets (SocM and WiFi), for a total of four models. Evaluation for all approaches was performed against subsets of the complete datasets kept separate specifically for validation purposes, and included six days in March 2016. Using each forecast model, a set of values was calculated for each validation date for the full day, at a quarter-hour resolution (96 predictions per day per model). The predicted values were then compared against recorded values for that date and time, and an error score was calculated using the symmetric mean absolute percentage error (sMAPE) metric. A percentage error was chosen here to enable comparison between datasets of different sizes. A full set of error statistics and error plots is offered in Appendices C.2, C.3, C.4, and C.5.

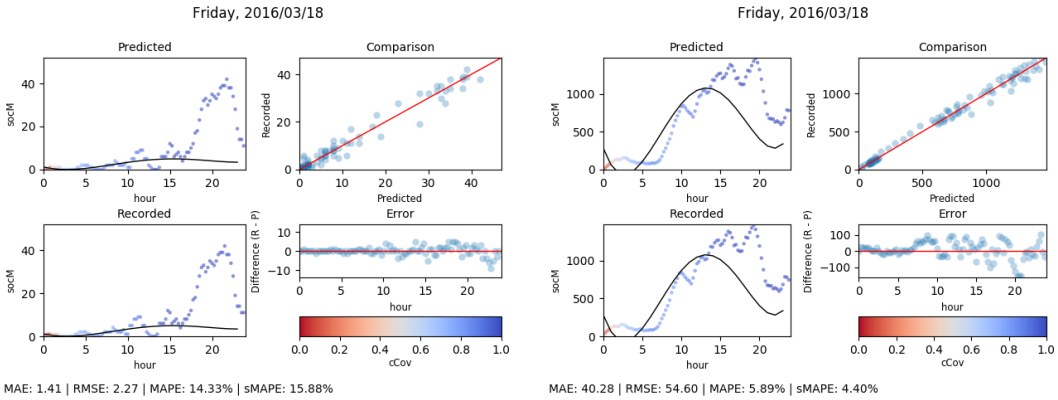
Date	SocM-Weather	WiFi-Weather	SocM Naive	WiFi Naive
2016/03/03	76.48%	31.91%	8.59%	6.89%
2016/03/07	32.84%	30.09%	10.26%	6.89%
2016/03/13	58.10%	43.76%	9.78%	7.66%
2016/03/18	46.08%	34.10%	15.88%	4.40%
2016/03/22	39.85%	44.60%	10.92%	7.66%
2016/03/29	49.32%	35.82%	16.54%	6.38%

Table 9.7: Forecast Models Validation for CS2:QEOP - sMAPE Values

As can be seen in Table 9.7, the naive model outperformed the linear models by a large margin across both datasets. Multiple reasons for this have been identified: Regarding SocM data, as has been discussed already, in the case of QEOP overall data volume is quite low, with instances were no SocM events are recorded even at busy times, and with no apparent dominant daily patterns. It was therefore expected that this method would not perform adequately. Regarding WiFi data, although the dataset demonstrates some degree of consistency across different days, total sample size was small (31 days), and therefore the forecast model was not adequately calibrated.



(a) SocM-Weather **(b)** WiFi-Weather



(c) SocM Naive **(d)** WiFi Naive

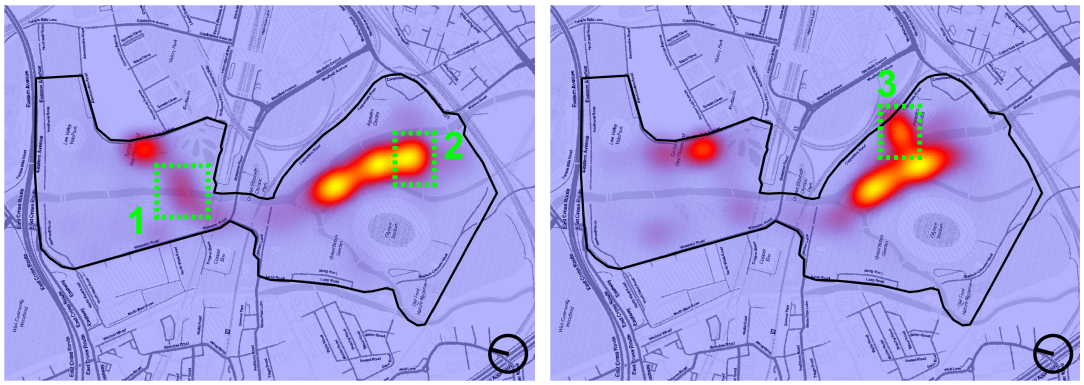
Figure 9.22: QEOP Activity Forecasts for 2016/03/18

In contrast, the naive model performed better due to the temporal fidelity of the dataset (quarter-hour records) which has been found to be well under the average park visit duration, and therefore values are not expected to change drastically between consecutive time-steps. A comparison between the four different forecast models for a sample day (2016/03/18) can be seen in Figure 9.22.

9.5.2 Spatial Disaggregation Model Evaluation

The Spatial Disaggregation Model (SDM) used in this case study implemented an ABM of park visitor activity, and was calibrated to capture activity in QEOP using actual visitor locations and activities captured via site survey. An evaluation of the accuracy of its spatial disaggregation was performed against a different visitor activity dataset captured at a different time. The evaluation process did not aim to

measure temporal accuracy or overall aggregate activity (these aspects are outside the scope of the SDM, and were captured in the forecast models), but rather to measure only the degree to which the ABM dispersed activity accurately throughout the park. For the synthetic population the model was run for a complete simulated day, and a record of all agent locations and activities was captured at a point in time when total agent population was comparable to visitor volume during the validation site survey dataset.



(a) Simulated (1197 agents) **(b)** Recorded (1520 visitors)

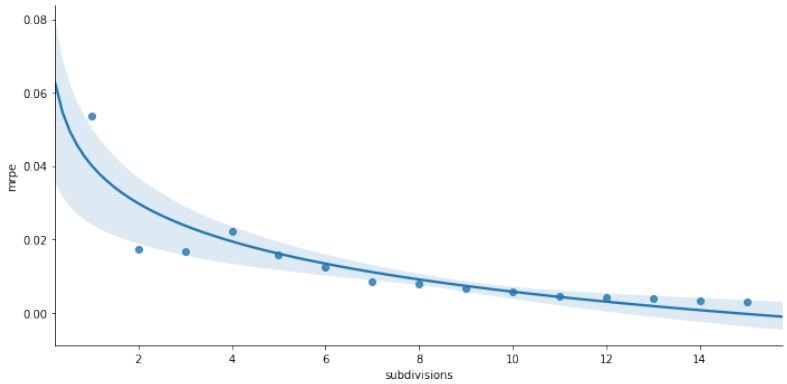
Figure 9.23: QEOP ABM Activity Heatmaps

An initial visual comparison of SDM results and actual activity can be seen in Figure 9.23, which demonstrates that the model accurately has generally succeeded in capturing the major hotspot locations. Some locations have been misrepresented in the model, and these have been marked in the figures (9.23a and 9.23b). More specifically, in location 1, the model has overestimated the amount of visitors crossing a bridge in the north part of the park, although it has not overestimated activity taking place at either side of the river at that point, but rather just the crossing of the river. In location 2, the model has extended the simulated activity further to the south than observed activity. Location 3 marks the most popular entrance to the park by far, arriving from the nearby shopping centre and train and underground stations, where activity has been extremely underestimated in the model. This is due to the fact that the particular location has been marked as an entry/exit point in the model, i.e. it is the location at which agents spawn and are removed. Given that agents do not begin their lifetime in the simulation in a stationary activity, it is

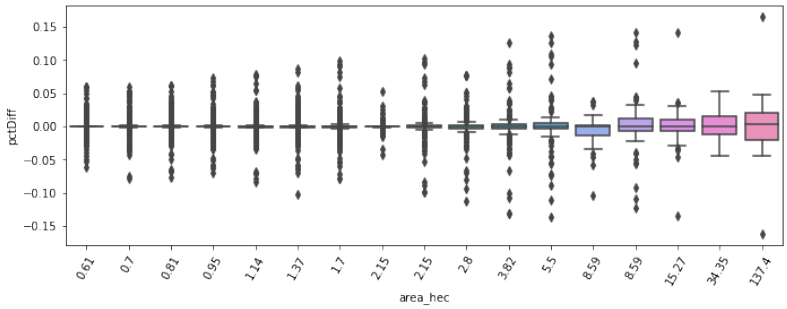
more than likely that they will have moved from that area by the time they engage in their first stationary activity. In contrast, actual visitors have already walked some distance from the shopping centre by the time they arrive at this location, and are more likely to rest before continuing in the park.

Apart from the 3 locations noted here, the model accurately captures the distribution of activity. To measure model accuracy, the *Expanding Cell Validation Method* was implemented. A grid was generated that completely covered all of the data points. It was then duplicated and moved 25% of a cell length in each of the cardinal directions, so that a total of 5 grids were created. The number of agents and visitors in each grid cell was counted as a percentage of the total population size for the respective dataset, and the difference between relative counts was calculated as the *relative percentage error* of each cell. Finally, the mean of all cell error values was captured as the overall Mean Relative Percentage Error (MRPE). This process was performed for multiple grid cell sizes, to see how the model performs at different scales, with grid cell sizes ranging from 0.61ha to 137.4ha, the latter completely containing all of the data points in a single cell. The code used to generate the grids and calculate the cell error is presented in Appendix A.5.2. A graph of MRPE at different measuring scales can be seen in Figure 9.24a, and the spread of cell error values by measuring scale is presented in Figure 9.24b.

As can be seen in in the error graph and plots of the error grids (Figure 9.25), error magnitude correlates with grid size. Similar to CS1:HyP, the validation scale for the model was defined by the observation scale of the site survey data, and so a grid cell size of 2.8ha was chosen as the cell size best matching an observation area of 3.14ha (as derived from a search radius of 100m). The validation grid is shown in Figure 9.25, and it can be seen that at this scale, it highlights the areas discussed previously (Figure 9.23), with a MRPE of 0.8% and a maximum relative error of 11.26%.



(a) MRPE by Subdivisions of Validation Grid



(b) Cell Error Distribution at Measuring Scale

Figure 9.24: QEOP ABM Spatial Error at Measuring Scale

Figure 9.25: QEOP ABM Spatial Validation. Error is measured as the difference between proportional grid counts (*recorded* − *simulated*). Red hues show model overestimation, blue hues show model underestimation.

9.5.3 Overall Model Evaluation

In addition to evaluating the individual components of the real-time modelling methodologies, CS2:QEOP aimed at evaluating the overall real-time model as well. This step investigated the accuracy of predicted spatial distribution of activity in QEOP *at specific times*, in other words it aimed to evaluate the full spatio-temporal properties of the real-time ABM developed in this work.

This validation step was not possible in CS1:HyP due to dataset limitations, as real-time data on visitor activity in CS1:HyP was captured via SocM, which did not include detailed spatial information. For CS2:QEOP, the WiFi dataset provided this potential, as wireless connectivity records include spatial information as well. This spatial information comes through appending to the dataset the individual access point which recorded an event, whose locations are known beforehand. Furthermore, the access points have a stated effective range of approximately 100 meters, and therefore it is possible to infer spatial activity in the area around each access point by the number of recent events recorded at that point.

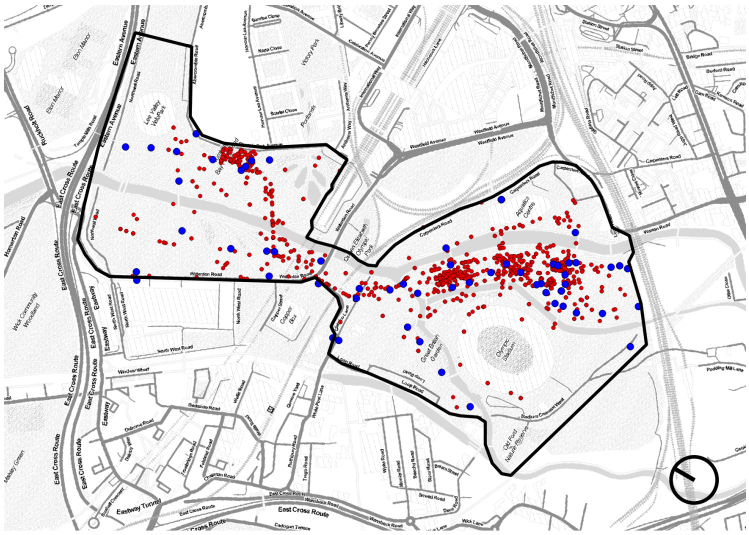


Figure 9.26: QEOP Simulated Visitor Locations (red) and WiFi Access Point Locations (blue)

As a first step in the evaluation of the overall real-time model, model results were matched in form to the validation dataset, to enable comparison. This was done by running the simulation for a full simulated day and extracting the locations of all

agents at the specific point in time that constitutes the selected validation datetime. These agent locations were then appended to the location of the closest access point (Figure 9.26), and the sum of all connections at each access point was counted as the simulated population at each access point. This value was compared to the hourly count of recorded WiFi events at each access point, considered here as the observed population. A visual comparison is presented in Figure 9.27.

(a) Simulated (1130 agents) **(b)** Recorded (1900 connections)

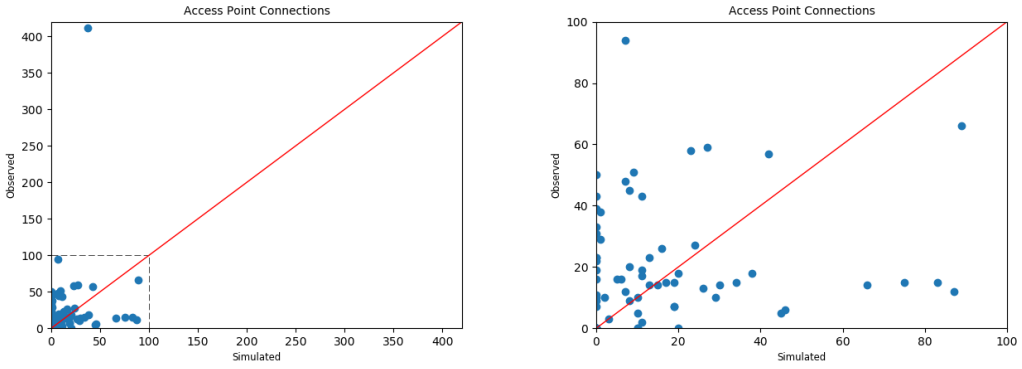
Figure 9.27: QEOP Activity Heatmaps at Access Points (2016/03/18 14:30)

It is evident from the heatmaps that model results do not correlate with WiFi connectivity records. In the simulation, connections at crowded areas are evenly distributed across all access points close to any area, whereas in the observations dataset most connections are recorded at only a handful of access points. Indeed, looking at the residuals plot of connections at access points (Figure 9.28), a single access point has 412 recorded connections (approximately a quarter of the total). However even disregarding that particular data point, it is evident that the simulation and observation data points do not correlate, as is confirmed when calculating the error statistics for the two datasets (Table 9.8).

Datetime	R^2	**MAE**	**RMSE**	**sMAPE**
2016/03/18 14:30	0.0163	25.83099	52.82312	55.32%

Table 9.8: QEOP Overall Model Error

Multiple reasons for this disagreement between simulated and observed results are considered. Regarding the assumptions in the preparation of the simulated dataset:



(a) Residuals - Full Range **(b)** Residuals - Constrained Range

Figure 9.28: QEOP Real-Time Model Results - Residuals Plot

the *nearest neighbor* method was used for aggregating agent locations to access points, which might differ from the way wireless networked devices connect to an access point. Regarding the validation dataset itself: A number of discrepancies were identified in the WiFi dataset, both to itself and to other data collected for the area. First, there is inconsistency between visitor activity recorded via site surveys (Figure 9.8) and activity recorded at access points (Figure 9.27); although both maps highlight the same locations, in the WiFi connectivity dataset a single location is over-represented. Second, the number of connections reported in the WiFi dataset appears to be overestimating, as it is comparable and at many cases higher than the number of people in the park, as recorded via site surveys. Further inspection of the WiFi dataset indicates that multiple consecutive connection attempts are often performed by the same device in a short duration at the same location (potentially while the device is negotiating connection to the network), therefore inflating the reported events at access points. Further to that, many connection sessions do not end with a final *disconnect* event to signal the definitive end of the session, meaning that it is not possible to attach beginning and end time to a session and thus calculate whether a device was actually in the park at a given point in time (and therefore more accurately calculate the total number and locations of devices in the park at a given point in time). For the reasons discussed here, it was decided that the WiFi dataset as was made available was lacking in veracity, and could not be used as a proxy for

visitor activity in real-time.

9.6 Summary

CS2:QEOP proved to be a successful case study overall concerning its exploratory elements. However, during the incorporation and subsequent testing of existing and novel datasets, some of them were found to be inadequate for the purposes of this work, at least during the time this research was conducted. Specifically, two of the initial aims were not achieved, due to limitations in the datasets: inferring real-time overall activity via SocM datasets, and obtaining a reliable high fidelity indicator of activity via a novel exclusive dataset (WiFi). All other objectives are considered to have been successful. More specifically, revisiting the case study aims as defined in the beginning of this chapter:

1. **Validation of CS1:HyP methodologies.** The successful Real-Time Public Space Activity Modelling approaches developed in CS1:HyP were applied again in CS2:QEOP to explore their validity. These were: capturing of SocM and weather data in real-time, estimating total visitor activity in the park, spatially disaggregating total activity using an ABM, and evaluating aggregate activity forecast and SDM. SocM adn weather data collection was performed successfully, however the returned volume of SocM activity was quite low, to the degree that forecasting activity was not feasible. This was made evident by successfully implementing an evaluation analysis similar to the one in CS1:HyP, which illustrated high error scores compared to CS1:HyP and other forecast approaches. The SDM was successfully implemented and calibrated to capture activity in QEOP, and the evaluation process using the *Expanding Cell Validation Method* further highlighted some interesting aspects of the spatial characteristics and clustering of activity in QEOP.

2. **Exploration of the potential of novel datasets.** CS2:QEOP made use of some exclusive datasets to capture activity in real-time, namely wireless con-

nections to the park's WiFi network. This dataset was successfully used to forecast overall activity in cases where forecasts using SocM were infeasible. At finer scales however (both temporal and spatial), issues with the veracity of the dataset became apparent, as the way data was recorded made it infeasible to infer the number of visitors in the park. Nevertheless, this was found to be not an issue with the dataset and the overall approach itself, but rather with the way data is stored at the moment, and with some changes it could indeed prove to be a useful tool for real-time crowd dynamic monitoring.

3. **Verification of PSA ABM capabilities in more complex environments.** The ABM used to simulate individual visitor activity in QEOP was overall successfully implemented in CS2:QEOP. The more elaborate geometries and landscape presented by QEOP were successfully handled by the ABM.

Chapter 10

Discussion on Case Studies

This chapter is devoted to a detailed discussion of results and findings around all areas of interest of this work, as identified through their application in the two case studies. Although some comments have been offered already regarding datasets, methodologies, and results in the two previous chapters, an in-depth discussion was deliberately withheld until this point. The reason for this was to discuss and review the findings not in the context of each particular area or case study, but rather in light of the endeavour to develop *Real-Time Simulations of Public Space Activity*, which is the overarching aim of this work. Furthermore, it was deemed necessary to discuss not only the results from the studies, but the methodologies and processes of the studies themselves. As such, this chapter will discuss overall results and findings of this work, grouped by appropriate thematic: Real-Time Data, Agent-Based Models of Public Space Activity, Real-Time Simulations of Public Spaces, and finally the case studies themselves.

The chapter begins (section 10.1) with a discussion on Real-Time Data (RTD), an essential component of Real-Time Simulations, as it was encountered in this work. This section will conclude the RTD thread running throughout this thesis, which was first introduced and defined in chapter 4, placed within the real-time modelling framework in chapter 5, captured in chapter 6, and applied in the two case studies (chapter 8 and chapter 9). Aspects of availability will be discussed first, as this was

established as a main characteristic of RTD (section 4.3). The informative power of the different datasets used in this work will also be discussed, considered here as their potential in capturing Public Space Activity (PSA), along with issues when used in real-time forecasting.

The next section (section 10.2) will discuss the models developed here that capture human activity in public spaces. It offers a conclusion on the remaining two conceptual threads of this work, the use of public space, and Agent-Based Models (ABMs), developed throughout this thesis in chapter 2, chapter 3, and chapter 7. This section will discuss models both in terms of behavioural rules and heuristics that formulated the final models, as well as model performance.

Concerning the real-time nature of this work, section 10.3 discusses findings relating to the overall development of *Real-Time Simulations of Public Space*. It will reflect on the methodologies developed here regarding their overall validity, and will furthermore consider their limitations and extensibility.

Finally, section 10.4 will discuss the case studies themselves. The focus here is on highlighting problems, findings, and potentials regarding the methodology and its application to a real-world examples, as identified through the course of the two case studies, Case Study 1: Hyde Park (CS1:HyP) and Case Study 2: Queen Elizabeth Olympic Park (CS2:QEOP). It will discuss the rationale behind area choice, characteristics of each area, as well as activities observed in each area, and will reflect and review on how all these aspects affected or were captured in the simulation of each space.

10.1 On Real-Time Data

This first section will focus exclusively on the Real-Time datasets used in this work. It presents a discussion on their primary properties, as well as any analytical work that stemmed directly and exclusively from the real-time nature of the datasets, in other words this section discusses the merits of RTD regarding their *Real-Time* na-

ture. This work explored the potential of multiple datasets, over a large period of time, gathering a year's worth of data. An overall tendency of this work is acknowledged here, in leaning towards the open and publicly available end of the spectrum of data availability.

10.1.1 On Data Availability and Informative Potential

As was established in the review of Real-Time Data (RTD) (chapter 4), two main characteristics have been identified regarding the nature of RTD: Temporality, and Accessibility. Temporality, or the time difference between an event taking place and the data point capturing it being generated and stored, was considered to be a given for any dataset used in this work, and will not be discussed here. Accessibility on the other hand (and subsequently reliability) was found to be equally as important to RTD, since if a dataset is not made available as soon as it is captured (for any reason), then it can no longer be considered as *real-time*. Therefore, a discussion on dataset availability will be presented here, for all real-time datasets ultimately used in the two case studies.

Publicly available datasets were found to be varied in terms of availability: Social Media (SocM) data was consistently available and reliably captured (barring some instances of script execution failure on the part of the researcher), up until drastic/important changes in some data sources' Application Programming Interface (API) terms and conditions made data capturing impossible, and in other cases changes in the API made it difficult to consistently capture data in an automated fashion. Weather data was found to be consistently available. Finally, the exclusive dataset tested here (WiFi) consisted of sample data, covering non-consecutive months, and was made available after said periods, i.e. was not real-time under the definition used here (*'RT-pub'*, as defined in section 4.1).

As can be seen in Figure 10.1, only two datasets proved to be reliably available throughout the duration of this work: Weather data, and Twitter data. Furthermore, there appears to be a correlation between the openness of the dataset (ranked based

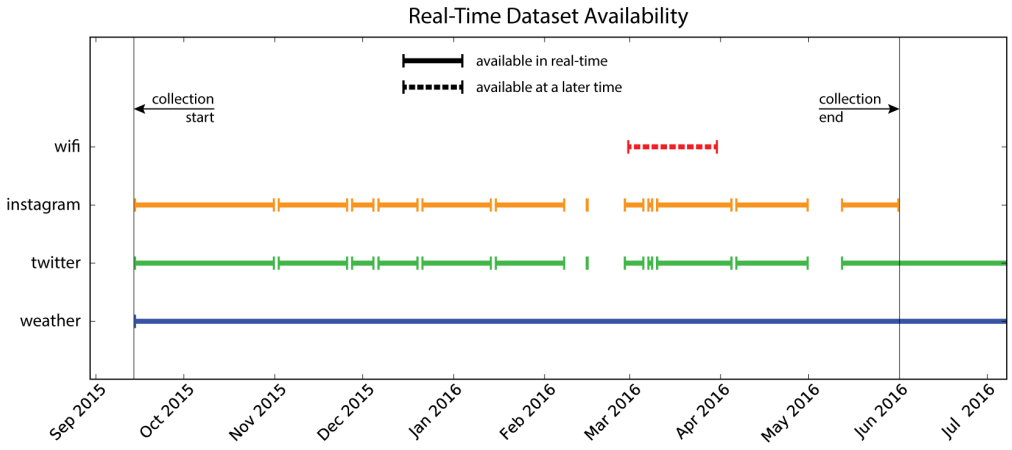


Figure 10.1: Real-Time Dataset Availability, by Dataset

on the Open data Institute's definitions, see Broad, 2015), and its overall real-time availability, at least for the datasets examined here. Weather data was sourced from services[1] (for ease of access) which themselves aggregate from multiple open and publicly available meteorological services, and is considered if not open, at the very least publicly shared data. SocM data (Twitter and Instagram) was considered as attribute-based access shared data, and therefore constrained access, given the requirements of setting up a developer account at each platform and fulfilling criteria regarding data collection. Finally, WiFi data was considered as named access shared data, as it was shared to specific researchers by the data provider themselves.

Given the potential ubiquity of the various datasets originally considered in this work, the overall informative potential of each dataset was also taken into consideration, meaning the potential each dataset had at capturing multiple aspects of the system being examined. For this reason, in addition to the temporal aspect of RTD, which was considered a given, the inclusion of any spatial information in the datasets was also considered, for capturing the distribution of activity in the areas of interest via proxy. At the conclusion of this work, no single dataset was found to fully exhibit both spatial and temporal aspects at an adequate degree to be used in real-time simulations of PSA. Most datasets (Weather, SocM, WiFi) provided data in a streaming fashion with timestamps, and therefore had a strong tempo-

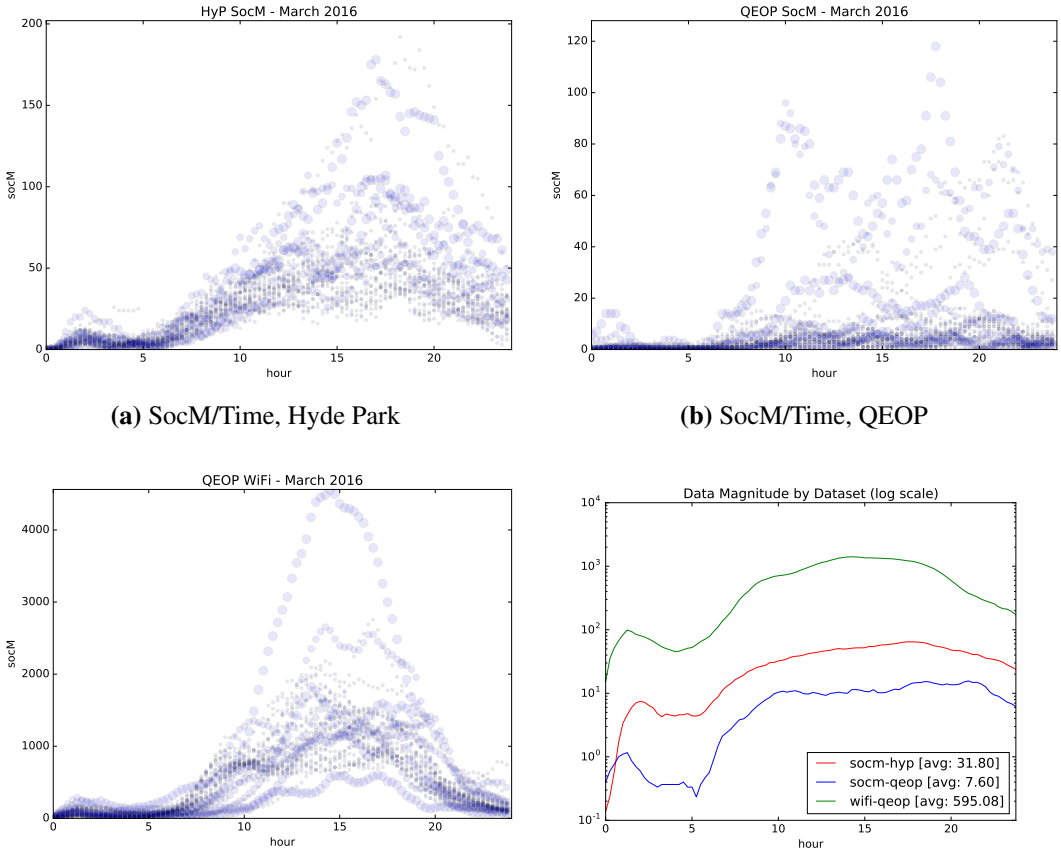
[1] darksky.net

ral presence, however, all of them lacked high-fidelity spatial information. More specifically, spatial variation in weather conditions was meaningless at the scale of observation. SocM data sources aggregated the data point location to the nearest significant location (as delivered through their respective APIs, e.g. Twitter often appended tweets originating from Hyde Park to the centre of the park regardless of original geolocation), and therefore lacked spatial fidelity. Finally, WiFi data did include spatial information at adequate detail, by recording the access point ID (of which the locations were known) for each event, however the dataset proved to be too noisy to use effectively.

10.1.2 On Modelling & Forecasting Capabilities of RTD

As was expected due to the statistical nature of the forecast models developed here, forecasting using RTD is more accurate when high volume datasets are available. This was observed in the two case studies as well, when comparing SocM data in each: the large volume of SocM data in CS1:HyP made forecasting using a Generalized Linear Model (GLM) possible, whereas in CS2:QEOP overall SocM volume in the park was low, and as a result there was much more noise, making it impossible to forecast accurately (Figure 10.2). Comparing the two case studies then, and considering CS1:HyP as a successful implementation of a real-time forecast model using SocM, some minimum values of SocM activity for forecasting PSA can be estimated: a daily average of 31.80 SocM per 15 minutes, with a consistent peak hours average of approximately 50 SocM per 15 minutes.

CS2:QEOP highlighted another important aspect of forecasting RTD pertaining to human activity at short intervals: Naive forecast models proved to be much more accurate compared to linear regression models. This is attributed to the ephemeral nature of the spaces examined here, which can exhibit rapid changes in volume of activity during a short time, as well as the quality of the datasets, which might introduce a significant amount of noise. However, even though naive models seem to outperform more formalized models, a note needs to be made here regarding the



(a) SocM/Time, Hyde Park

(b) SocM/Time, QEOP

(c) WiFi/Time, QEOP

(d) Dataset Magnitude (March 2016 average)

Figure 10.2: Dataset Volume Comparison

nature of both approaches: naive forecast models such as the one implemented in CS2:QEOP *require* RTD as input to produce *any* output. In other words, naive models are reliant on a constant stream of data that delivers the 'predicted' variable at a delay of one timestep, and are therefore reliant on a single point of failure. As has been presented, this stability in data availability is not guaranteed, at least for the time being, and at least for the datasets examined in this work. Additionally, naive models introduce inherent error, by definition. GLMs (or other correlation-driven forecast models) on the other hand have the potential to provide more accurate results, given good-quality calibration data. Furthermore, as they rely on predictive input variables to output predicted values (e.g. SocM-weather correlation in this work), they can still function in cases where predicted data is missing, in the case of an outage for example, as long as the predictive datasets are fairly reliable (as

seen in this work in the case of weather data, compared to SocM data). Regarding then the overall forecasting capabilities of RTD as seen in the two case studies, this work concludes that at this point, RTD does not seem to hold adequate veracity as a whole to support meaningful, reliable short-term continuous predictions on urban public space activity at the scale of the individual.

10.2 On Agent-Based Models of Public Space Activity

One of the main research questions posed in this work was whether state of the art tools could support the development of a perpetual, real-time simulation of a city's public spaces, one of the principal components of which is the human interaction that takes place daily within them. In investigating this question, this work developed a modelling framework capable of capturing and simulating human activity in public spaces using the Agent-Based Modelling paradigm, by translating the findings of Public Space Use studies into behavioural rules. In this regard, one of the aims of this work was to extend existing *streetscape* models (Torrens, 2016), i.e. pedestrian and crowd movement ABMs, into *parkscapes*, ABMs capable of capturing and simulating the wide range of activities taking place in public spaces. The resulting model was applied in two real-world scenarios, and was found to be overall successful. In this section, the specific shortcomings, limitations, and assumptions that went into the development of this *Agent-Based Model of Public Space Activity* will be discussed.

10.2.1 On Human Behavioural Characteristics

In developing the public space activity model, this work reviewed and codified observations on how people act and interact in public spaces (section 2.2, section 2.3, chapter 7). During this process, a number of assumptions were made, that resulted

in a simplified model of PSA[2] regarding individuals' behaviour. This simplified approach was chosen for multiple reasons: First, this thesis followed a *reductionist* approach, whereby a conscious attempt was made to apply simpler solutions where possible, for a variety of reasons (model legibility and comprehension, verification and validation, extensibility). Second, given the scale, size, and scope of the areas studied here, it was assumed that a simplified model of human behaviour would be more than adequate in capturing crowd dynamics. More specifically, as the Environment/System scale is for a whole park, Agent/Component scale is the individual visitor, and more importantly the objective of the model is to simulate *overall* dynamics, a simplified model would hold more explanatory value, and furthermore could be calibrated using available data.

10.2.1.1 Evaluation of Movement Heuristics

Wayfinding and navigation was implemented using shortest-path algorithms and angle-constrained random walks. These heuristics have been discussed to great length in literature, argued for and against (as presented earlier, in subsubsection 2.2.1.1). Given the characteristics of the spaces chosen as case studies, in the extent of this thesis the selected navigation algorithms are considered to adequately represent the behaviour of the target population. More specifically, parks do not attract overly intensive movement and transit activities, are in open environments, without a particular goal/destination in place, therefore a wandering behaviour can be assumed to capture the wayfinding behaviour of park visitors. Furthermore, due to the mostly open terrain, visitors have an unobstructed view of their surroundings, and therefore if a particular attractor/goal is identified, a direct path will almost certainly exist to that destination. In this regard, and according to relevant literature (Gehl, 1987, Gärling and Gärling, 1988, Whyte, 1988, Bitgood and Dukes, 2006, Jazwinski and Walcheski, 2011), the shortest-path algorithm appears to adequately capture small scale path-planning in park visitors. However, wayfinding in this

[2]This was of course expected from the outset, as urban and spatial models constitute a simplification of the system they represent (Batty, 2001).

model can certainly be improved, at multiple scales. In the larger scale: extensive discussions on path-planning are found in the fields of Psychology and Behavioural Geography, where additional valid approaches and heuristics to human wayfinding are presented (as listed e.g. in Spiers and Maguire, 2008). Such approaches can certainly be tested in simulations and models as presented in this work, as their implementation can provide a more realistic simulation of human wayfinding behaviour in open spaces. In the smaller scale: In this work, small scale movement and crowd dynamics such as obstacle avoidance were largely unexamined. Recent advances in computer vision research (Sprague et al., 2007, Ondej et al., 2010) and in the simulation of locomotion dynamics such as inverse kinematics (Tolani et al., 2000), as well as overall crowd pedestrian simulation (Guy et al., 2010, also see overall field review in Torrens, 2016) can provide a clearer image of these aspects of crowd interaction.

10.2.1.2 Evaluation of Social Interaction, Crowding, and Stationary Activities

Regarding the implementation of crowding dynamics in stationary activities, in relation to social interaction and crowding distances. The primary point to be made here is certainly the inclusion of repelling activities. First discussed in *Chapter 2: Understanding Public Space Use*, it was considered that all social interaction in public spaces is considered as a positive feedback interaction. Although the opposite is known to happen as well, it was not included for brevity, as it has been observed that social interaction is overwhelmingly a positive interaction (Jacobs, 1961, Whyte, 1980, Gehl, 1987), even if considering the passive aspects (e.g. people-watching).

Additional (short) discussion on potential model over-calibration, regarding fixed-point attractors ('features' in the ABM, used by agents when in a 'Feature Visit' state). These elements in the model potentially introduce an 'over-fitting' aspect, as they constitute fixed points, known to attract large crowds in observed activity,

and are included in the model explicitly by location. In other words, the model predicts increased activity in a specific location by explicitly directing entities to visit that location. On the other hand however, in both case studies, these fixed locations of increased activity *are* known beforehand, and *are* expected to exhibit increased activity, as they constitute main attractions in the area, and oftentimes their existence relies on such increased activity (in most instances they were found to be restaurants, or public discourse locations). It may be argued therefore that they might indeed be an 'anomaly' to the otherwise expected undisturbed activity distribution, and their accurate simulation requires them being set explicitly.

10.2.2 On Agent-Based Model Performance

Model performance is considered here from two distinctly different perspectives: First, on the model's stated objective, and more specifically on whether the conceptual model performed well in capturing individual park visitor behaviour, and its implementation in simulating the distribution of activities in the area of interest. Second, on how the model performed computationally, or whether this approach constitutes a viable computational technique in simulating public space activity. Although the two perspectives presented here can be considered independently, they appear to be interconnected to some degree in the scope of the models developed here. This is due to one of the requirements set earlier in this work, of developing models of human spatial behaviour capable of functioning in a fully three-dimensional environment. Therefore, while the addition of the third dimension enables a more accurate representation of the space of interest, at the same time it has a negative impact on the computational performance of the simulations.

Regarding the model's computational performance, simulations were run on two different PCs, one Alienware desktop machine[3], and one MSI laptop machine[4]. A pair of simulations of CS2:QEOP was run on each machine, with an increasing population starting with 500, increasing by 500 at each controller update, with a

[3]CPU: i7 @3.00GHz, RAM: 32GB, GPU: 2x NVIDIA GTX980Ti
[4]CPU: i7 @2.40GHz, RAM: 8GB, GPU: NVIDIA GTX850M

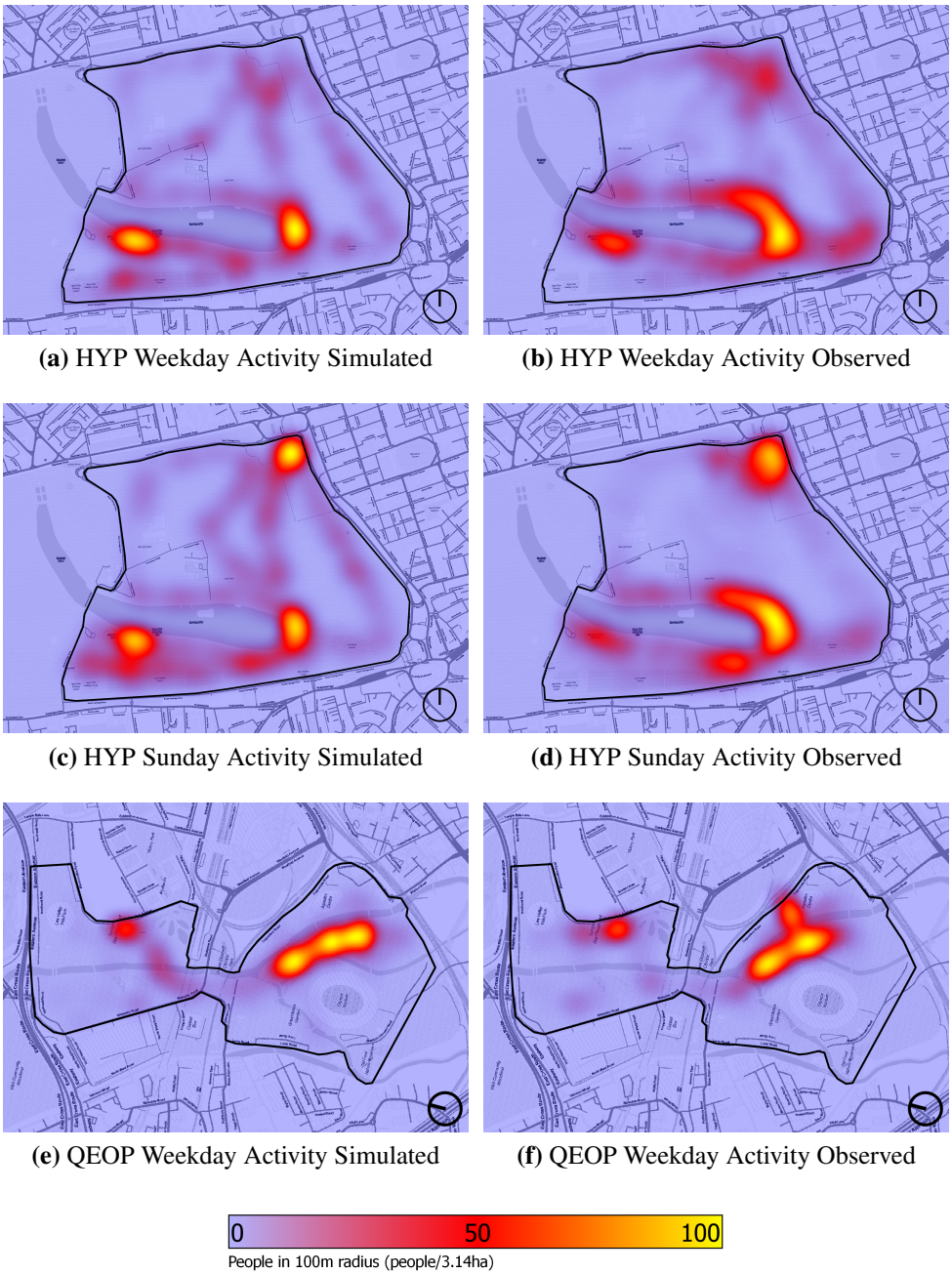
cap on 2000, and the delay between frames was recorded. Given that the model is in a 3D environment, rendering poses a bottleneck; by disabling rendering, the FPS noticeably improved in both machines. A comparison of model performance between the two machines is shown in Figure 10.3.

Figure 10.3: ABM Computational Performance

Regarding the model's performance on the stated objectives, results from the two case studies suggest that the model performs well overall in capturing and simulating public space activity. In both case studies, locations of the main hotspots of activity were accurately predicted, as well as locations of minimal to no activity ("cold spots"), as can be seen in Figure 10.4. That said, the model exhibited an overall tendency to under-represent the difference in activity volumes between high and low activity locations, i.e. the model tended to under-represent hotspots and over-represent cold spots. Over-representation of cold spots was attributed to the stochastic nature of the model, which imbued agents with a random wandering behaviour. Under-representation of hotspots was attributed to the fact that most hotspots were identified as being features and attractions (e.g. restaurants, playgrounds, etc.), which while included as a behaviour in the model, was not calibrated for properly in terms of visitor volumes[5].

Regarding the agent behavioural framework developed in the model, observations

[5] And was not meant to, as no dataset was found that captured equally and in detail visitor volumes for all the different locations identified as features in the two areas.



(a) HYP Weekday Activity Simulated

(b) HYP Weekday Activity Observed

(c) HYP Sunday Activity Simulated

(d) HYP Sunday Activity Observed

(e) QEOP Weekday Activity Simulated

(f) QEOP Weekday Activity Observed

Figure 10.4: Case Study Heatmaps - Simulated vs Observed

and findings from a range of different studies of public space use were reviewed, and as has been mentioned earlier, ultimately development settled on a straightforward classification of visitor activities into two broad categories, namely moving and stationary activities. This was assumed to provide a good overall description of activities with both brevity and detail when observed at the scale of a whole ur-

ban park. Given the nature of the public spaces examined in this work, empirical evidence suggested that: a. there would be people moving through them, as public spaces are the de facto transit space, and b. there would most likely be people engaged in stationary activities in them, as has been observed in previous studies (Whyte, 1980; 1988, Gehl, 1987). This initial assumption on the broad classification was confirmed in external surveys (Ipsos Mori, 2015b), which included a wide variety of activities as the purpose of visit in park visitor responses, as well as site surveys conducted in this work (section 8.2, section 9.2), which observed visitors engaged in multiple types of activities, both involving movement and stationary, including restaurant visits, sports, exercise, walks, among many others. All the different activities were classified under the two general categories of 'moving/stationary', as producing an exhaustive list of activities would require a quite lengthy list, and furthermore would require extensive calibration in order to verify that they accurately represented the actual activities taking place in the park. In addition, no real-time dataset was found that tracked all of the individual visitors' activities in detail throughout their visit, and therefore the scope of potential activities could not be identified.

One part in which the activity classification went into further detail however was a sub-categorisation of stationary activities, into general sitting, sports, and feature/attraction visit. These three types were generally observed during visits in both case study areas, and it was felt that they captured the full range of activities taking place in a park, as each expressed different model mechanics which encompass a range of specific activities. More specifically, feature visits stand for a direct and purposeful route and stay at a specific and predefined location, do not necessarily involve interaction between agents, involve minimal interaction between agent and environment, and may include activities such as sitting at a restaurant/cafe, visiting an event, visiting a tourist attraction/monument, etc. Sports activities stand for any activity that requires a specific set of conditions be presented by the environment[6],

[6]In this work the conditions were that an area was clear of obstacles such as trees and buildings, was fairly level/did not have intense landscape, and was clear of paths

and requires some interaction between agents, either attracting or repelling, based on agent type. They are therefore not location specific, and agents could engage in this type of activity at any location that fulfilled all conditions, they exhibited therefore a foraging behaviour in looking for an appropriate location. Finally, general sitting activities stand for all other stationary activities in public spaces, which according to literature (Whyte, 1980; 1988, Gehl, 1987) are dependent on the existence of other agents in the area, and constitute positive-feedback loops in the model.

Although not captured explicitly in the surveys, these activity types were observed and therefore applied in the model under the assumption that given proper calibration, they would improve model accuracy. Indeed, as was shown in CS2:QEOP, calibrating the probabilities for each resulted in a reduction of overall model error. Furthermore, the interaction between activity probability, activity duration, and agent lifetime provided a large degree of agent heterogeneity, while keeping overall activity type distribution within expected value ranges. As can be seen in Figure 10.5, for agents with similar lifespans, the respective activity profiles over time differ greatly.

Figure 10.5: Heterogeneity in Agent Behaviour: Activities over Lifetime

A final point of discussion regarding the ABM developed here is to be made on its

spatial dimensions aspect, as all agent behaviours developed in this model were made to function in a fully three-dimensional environment. As has been discussed already, this introduced a disadvantage in terms of computational performance, but offered increased detail and descriptive capability, compared to a two-dimensional or pseudo-3D implementation. A 2D implementation for the agent framework was considered, and indeed it would potentially have been more applicable for CS1:HyP: Hyde Park does not exhibit any intense relief in terrain, and all of its activity can be assumed to generally take place on flat terrain, therefore a 2D implementation would have been applicable and more computationally efficient. CS2:QEOP on the other hand includes overlapping geometries, hills, and noticeable terrain relief. Although activity could have been simulated in 2D, doing so would have required a set of assumptions and abstractions regarding agent cognition and behaviour regarding the third dimension.

It was decided that both case studies should be studied within the same simulation framework. A three-dimensional implementation of the ABM was chosen, in CS2:QEOP for necessity, in CS1:HyP for testing purposes, and more importantly on both case studies, for reasons of descriptive capability: As has been discussed previously (Bonabeau, 2002), ABMs offer a natural description of a system composed of individual entities. In the same mindset, it was decided that a full 3D representation of the environment and the subsequent integration of agent interaction within this 3D environment would provide a more ”natural” and comprehensible simulation of urban space, and it was believed that state of the art computer systems, software, and development platforms were more than capable of supporting this. Results from the second case study do indeed suggest that a 3D model performed well in capturing activity in QEOP, and doing so did not introduce insurmountable computational load, therefore suggesting a valid approach. This then enables the simulation of a range of urban spaces which would not have been possible in a 2D implementation, spaces exhibiting more complex design approaches diverging from the archetypal open flat town square or plaza, often developed over multiple levels (see for example Figure 10.6).

(a) Southbank Centre, London

(b) Park Guell, Barcelona

(c) Rockfeller Plaza, New York

(d) Cabacera Park, Valencia

Figure 10.6: Multi-Level Urban Spaces

10.3 On Real-Time Simulations of Public Space Activity

As has been stated multiple times in this thesis, the aim of this work is to develop a *Real-Time Simulation of Public Space Activity*. When discussing the details of this aim (section 5.1), three particular objectives were identified, that the underlying model would need to achieve. Specifically, the model would need to:

1. Accurately predict the volume of human activity in a public urban space at high temporal fidelity.

2. Accurately predict the types of activities taking place in a public urban space *and* the locations of said activities.

3. Perform the aforementioned predictions of activity concurrently with it happening, i.e. *in Real-Time*.

The first objective was achieved to some degree of accuracy. As was discussed earlier in this chapter (section 10.1), this work investigated the potential of RTD in continuously capturing and forecasting activity in public urban spaces. The investigation focussed on forecasts at short time intervals of 15 minutes, and it was found that under certain conditions (specifically given data stream consistency, availability, and volume), overall aggregate activity in an area could be predicted using proxy datasets.

The second objective is considered to have been largely achieved. The Agent-Based Model (ABM) of Public Space Activity (PSA) developed in this work incorporated human behavioural characteristics in public spaces as observed in public space use studies, was calibrated to ground truth data, and was set up to run using input from the aggregate activity forecast model. As discussed in section 10.2, the resulting model was found to accurately capture the dispersion of activity in the areas of study.

The third objective as a whole was tested against available datasets, with mixed results. More specifically: on the one hand, predicting aggregate activity in real-time was conditionally successful dependent on data being available in large volumes, as it was shown (CS2:QEOP) that smaller volumes introduce proportionately large amounts of noise, thus reducing accuracy. On the other hand, predicting the spatial distribution of activity in real-time was considered to not have been thoroughly examined. More specifically, it was found to be possible, and indeed was implemented with a Spatial Disaggregation Model (SDM) of activity running at 60 times real-time speed, more than enough to continuously predict the spatial distribution of activity in real-time. However, no real-time dataset was found that captured spatial characteristics of park visitor activity at a fine spatio-temporal scale, and therefore the SDM was not validated in real-time. For this reason the overall third objective is not considered to have been answered adequately, as available datasets were found

to be unable to support it.

10.4 On Case Study Areas and Findings

This section will focus on aspects relating to the two case studies themselves, including their physical characteristics, findings, results, methodologies, and limitations. As will be discussed later in more detail, the overall aim was to investigate two similar cases, in order to minimize the effect of external variables, and allow for meaningful interpretation and comparison of results between the two studies. In addition to the many physical similarities of the two areas (both are public use parks of comparable size, with similar features), the two case studies followed a similar analytical methodology: ground truth data on visitor activity was collected via site surveys, both studies implemented the same ABM to simulate visitor activity, and real-time forecasts of aggregate activity were performed using a GLM in both cases. One major difference between the two cases relates to the data sources: CS1:HyP was carried out with the additional limitation of employing only publicly available datasets (*Open Data* and *Public Access Shared Data* in the Open Data Institute's terms (Broad, 2015)). CS2:QEOP employed data from all sources used in CS1:HyP, as well as additional exclusive WiFi connectivity data.

10.4.1 On Area Choice

This work focussed exclusively on Public Space Activity (PSA) taking place in parks allocated to public use. This choice of target areas for both case studies was done deliberately, for a number of reasons, first the public nature of parks, and second their openness of space. On the first point, the public nature of parks attracts leisure activities, i.e. non-necessary activities, and therefore, park visitors can be expected to be driven by their own preference, rather than obligation, for visiting the space. Furthermore, the public nature of parks suggests that visitors look for, or at the very least are aware of, the potential for social interaction with other park visitors, even at the passive level of being in the same area with others. Therefore, given

the public nature of parks, the overall activity observed in parks can be assumed to be deriving from the ”attractiveness” or ”appropriateness” of specific locations for specific activities, and from the interaction between visitors.

On the openness of park space, parks mostly offer a continuous plane for activity, which reinforces the element of interaction between visitors: often no divisions or obstacles exist between two locations in a park, and therefore some degree of spatial autocorrelation can be expected in the observed activity. Furthermore, the continuous, open area of parks enables visitors to move in potentially any direction without restriction. In contrast to parks, more urbanized plazas or even indoor areas were originally considered as potential case study areas. However, urban areas often involve an important factor heavily affecting the movement of individuals, that is the motorized traffic network, which is by itself a significant area of research. The possibility of extending one case study to include urbanized areas or even focus it completely on a fully urbanized location was considered, but ultimately it was decided that the two case study areas would be too dissimilar to enable any meaningful comparison. For the reasons discussed above, the two parks (Hyde Park (HyP) and Queen Elizabeth Olympic Park (QEOP)) were chosen as the two case study areas.

10.4.2 On Physical Characteristics

Figure 8.1: Hyde Park (repeated from page 210)

Figure 9.1: Queen Elizabeth Olympic Park (repeated from page 252)

As was mentioned, one of the aims in choosing appropriate locations for the case studies was for them to be similar in enough aspects so that meaningful compari-

son between results of the two could be performed. Therefore, in addition to their similarity in use and function, both being public-facing parks that attract leisure activities, the two case study locations share a range of additional, physical characteristics. Regarding their basic characteristics, as can be seen in Table 10.1, the two parks have comparable surface areas, however they differ in perimeter and shape, as Hyde Park (HyP) is fairly compact, while Queen Elizabeth Olympic Park (QEOP) presents a more elongated shape with a narrow pass at its middle (Figure 8.1, Figure 9.1).

Case Study	**Surface Area (km^2)**	**Perimeter (km)**
HyP	1.273	4.875
QEOP	0.9645	5.894

Table 10.1: Case Study Area Physical Characteristics

Further to basic physical characteristics, the two parks are similar in terms of features found within them. First of all, both have significant bodies of water running through them, although in the case of QEOP, the River Lea divides the park into islands connected by multiple bridges, whereas in the case of HyP the Serpentine River is wider and obstructs all movement between its banks, but does not completely run through the park. In terms of vegetation, both parks have areas ranging from thick woods and groves to open lawns. Furthermore, both parks have 2 restaurants each, which appear to act as significant attractors. Given the similarities listed above, any findings on PSA observed to take place in any of the two locations could be considered as indicative of public space activity in general, rather than exhibiting a nuance of the particular area it was observed in.

10.4.3 On Observed Activity in Areas

Considering activity in each area, there appear to be significant differences in the volume of visitors observed. As can be seen in Table 10.2, Hyde Park consistently attracted more visitors, almost double the number of visitors observed in QEOP, and even accounting for the size difference between the two parks, HyP has 1.4

times more users per km^2. Furthermore, the values shown here may be expected to be slightly skewed in favour of QEOP, considering the survey dates: For CS1:HyP surveying took place in late October, with favourable weather (few clouds). For CS2:QEOP surveying took place in late August, with very favourable weather (high temperatures combined with clear skies). Therefore, it is estimated that under identical conditions, relative visitor numbers may be found to be even higher in Hyde Park.

Nevertheless, the difference in visitor volumes observed here is attributed primarily to historic, cultural, and locational characteristics of the parks: HyP has existed for nearly 5 centuries, while QEOP was constructed for the London Olympic Games of 2012, and therefore it is expected that visitors of Hyde Park have established routines and activities that may span years and decades, whereas QEOP has potentially not been assimilated yet into the daily or weekly routines of its users. Furthermore, Hyde Park's proximity to Central London along with its connectivity may make it a more viable destination for people commuting to and from central London.

Case Study	Visitor Count	Visitors Stationary	Visitors Moving	Stationary Percentage	Visitors/km^2
HyP	4599.25	2457.25	2142	52.54%	3612.92
QEOP	2479.25	1352.25	1127	53.87%	2570.50

Table 10.2: Case Study Activity Comparison (Average of all surveys for each case study)

It is interesting to note however that apart from overall visitor volume, the two parks exhibit similar activity profiles. Of all recorded visitors, slightly more than half were observed to be engaged in stationary activities. Furthermore, the most crowded locations of both parks appear to be restaurants, cafes, and other fixed-location attractions (e.g. the *Speakers' Corner* and *Diana, Princess of Wales Memorial Fountain* in Hyde Park, the playground and water fountain in QEOP). Given the multiple similarities between the two parks, it is assumed then that visitor activity profiles as observed in the two case studies presented here might be indicative of general park visitor activity profiles, if not for urban parks in the UK, then potentially for parks in London.

Chapter 11

Conclusion

This concluding chapter presents a summary of the thesis. Opening aims and objectives are re-addressed via the work presented here, and findings, limitations, and shortcomings of the work are discussed holistically in more detail, and within the overall context. Finally, directions of future research are outlined.

11.1 Review of Aims & Objectives

At the opening chapter of this thesis, the aim of this work was stated (section 1.3):

> [...] to develop an *Agent-Based Modelling* framework of *Public Space Use*, calibrated using *Real-Time Data* streams, and applied to a simulation of current activity and conditions of Public Spaces; a *Real-Time Simulation of Public Space Activity*. (section 1.3)

This overall aim was furthermore deconstructed into individual objectives, each one addressing a different aspect of one of the three main fields (Public Space Use (PSU)/Public Space Activity (PSA), Agent-Based Models (ABMs), and Real-Time Data (RTD)), and through the completion of which the fulfillment of the overall aim could be achieved. The individual objectives were:

1. Review existing literature on studies of Public Space Use, and identify pre-

vailing hypotheses of public space user behaviour and rules of interaction.

2. Review spatial modelling approaches, and identify appropriate methodologies for modelling the activity of individuals in public spaces.

3. Review potential Real-Time Data Sources pertaining to activity in Public Spaces, and develop methodologies to capture and analyze selected datasets.

4. Develop a general framework for Real-Time Models of Public Space Activity.

5. Based on the outcomes of objectives 1, & 2, codify identified behaviours, build a spatial model of Public Space Activity, and couple with the general framework developed in 4.

6. Through the combination of objectives 3 & 4, couple the general framework model developed so far with Real-Time data feeds.

7. Apply the Real-Time Model of Public Space activity, and evaluate against real-world conditions.

This work addressed each of the objectives in a systematic approach. The first part of this thesis (Part I) devoted three chapters (Chapters 2, 3, and 4) to reviewing each field independently, and set the theoretical framework for the rest of this work. Objectives 1-3 were addressed in this part, by presenting each of the three main fields of interest (Public Space Use (PSU), Agent-Based Models (ABMs), Real-Time Data (RTD)) in detail, discussing aspects of each that were relevant to this work, and identifying connections between the three. More specifically, human behaviour in public spaces was presented through a review of previous studies on public space use, and findings were summarized into a codified set of spatial human behavioural rules. Furthermore, spatial computational modelling methodologies were reviewed in order to identify approaches relevant to this work, which were identified in the field of Individual-Based Models (IBMs). Within IBMs, the Agent-Based Modelling paradigm was finally identified as the most applicable for simulating public space users' activity, and was subsequently presented in depth. Finally, RTD was

reviewed within the context of Big Data, its merits and shortcomings compared to ”traditional” data collection methodologies were discussed, and properties of RTD relevant to activity in public spaces were identified.

In the second part (Part II), consisting of Chapters 5, 6, and 7, the methodologies for developing *Real-Time Simulations of Public Space Activity* were discussed in detail, thus addressing objectives 4-6. A conceptual model framework was presented first, described as a combination of two sub-models working in series: A forecast model of aggregate activity for providing continuous short term predictions of total activity in an area, and a Spatial Disaggregation Model (SDM) of individual visitor activity for translating forecasts into spatially distributed activity. The forecast model was then supplemented with RTD and thus the real-time nature of this work was implemented, and the SDM was developed and presented in detail using the Overview, Design concepts, and Details (ODD) framework, therefore completing the presentation of the *Real-Time Model of Public Space Activity* developed in this work.

The third and final part (Part III) presented the application and evaluation of the resulting model. Two case studies focussing on park activity and a discussion on their results were presented over the three final chapters (Chapters 8, 9, and 10), thus fulfilling the final objective, objective 7. Case Study 1: Hyde Park (CS1:HyP) mainly explored the validity of the overall method, concluding that the overall approach does hold merit, and the various simulation and RTD collection tools can indeed support the development of a Real-Time Simulation of Public Space Activity. Case Study 2: Queen Elizabeth Olympic Park (CS2:QEOP) extended the methodologies developed in CS1:HyP by applying the simulation to a different area, and offered an evaluation of the overall model. CS2:QEOP results suggested that an ABM implementing human behavioural rules in public spaces can capture and simulate park activity, however available data was was found to be inadequate in validating the overall model in real-time, and especially so regarding spatial aspects.

The aim of this work was to explore and examine the potential of state of the art

real-time data sources and simulation methodologies in supporting the development of a detailed, continuous, real-time simulation of the city. With this in mind, this work identified the most applicable modelling framework and developed a model of activity in public spaces, which was found to adequately capture spatial activity in parks. Furthermore, it developed methodologies for capturing real-time data pertaining to activity in public spaces, focussing mainly on publicly available datasets. Based on the analysis of the various real-time data sources captured, short-term predictive models of park activity were developed and evaluated, but overall were found to be unreliable in forecasting activity at short intervals accurately. Given all of the above, this work considers the overall aim of developing *Real-Time Simulations of Public Space Activity* to have been largely achieved: the components of the overall model were developed and individually evaluated successfully. One point that was not achieved however, is an evaluation of the overall Real-Time Model of Public Space Activity. The reason for this is that this work was not able to identify a single data source or combination of data sources that could offer a detailed record of spatial activity of individual park visitors in real-time. As such the spatial distribution of activity was not evaluated in real-time, and therefore the real-time spatial aspect of this work's overall aim was not explicitly answered.

11.2 Critique

This section discusses points of criticism on the work presented in this thesis.

On focussing on publicly available datasets As discussed in chapter 8, given the focus of this work on public spaces, it was determined that it would be of interest to examine the degree to which public physical life is captured and represented in our public digital traces. Also, the use of an exclusive (non-publicly available) dataset did not result in significantly increased model accuracy: in a direct comparison between Social Media (SocM) and wifi in CS2:QEOP, forecasts using wifi data did indeed preform much better than SocM. However when comparing R^2 scores in linear models of wifi-weather from CS2:QEOP with SocM-weather from CS1:HyP,

both datasets seem to perform similarly. Therefore, the exclusive dataset does not seem to introduce additional accuracy/information by itself when compared to publicly available datasets.

On not evaluating the overall model As discussed in the previous section, no dataset was found that contained high-fidelity spatial and temporal data, delivered in real-time. Therefore one issue that can be identified with this work is an initial overestimation of the capabilities of RTD overall, however the examination this work performed was potentially necessary to identify that at this point in time, RTD by itself (and especially publicly available RTD) is not capable of fully supporting the development of *Real-Time Simulations of Public Space Activity*, that capture PSA in high fidelity in both space and time.

On working in breadth rather than depth This work's focus was placed on the combinatorial potential of the three fields: PSU, ABMs, RTD. This work identifies the rapid changes in many fields around us, most importantly with the advent of the age of data and information, as well as sensing and IoT which generate immense volumes of data in real-time, but also in terms of computational power, which enable models and simulations to run at ever finer scales with increasing speed and efficiency. This work wonders then what impact these changes might have on our cities and the collective culture they express, most often seen through the interactions they mediate and facilitate in their public spaces, which also appear to be undergoing some significant changes, with most cities (as a public-ownership entity) literally losing space, to new developments, gentrification, and the POPS (privately-owned-public-spaces) phenomenon. Therefore, it was decided that in the extent of this work, it would be more crucial to explore the potential that these new tools and datasets might have in aiding in the understanding and shaping of current and future manifestations of public space, and for this reason focus was placed on examining the three fields mentioned above *in combination*.

On opting for 3D models It is acknowledged that calculations in three-dimensional space are more more computationally expensive than two-dimensional space. When

considering ABMs, 3D environments incur a significant computational load, and additionally, and potentially more importantly, they require more calibration and verification, due to the complexities the third dimension adds. It has been suggested therefore that when designing an ABM, the modeller needs to make a decision early in the design stage on whether to develop the model in 2D or 3D, as this will have a great impact on the rest of the development process. In this work, the decision was made early on to develop models of activity in public spaces in 3D environments. The urban built environment within which we humans move, act, and interact is predominantly perceived through its 3rd dimension, seen in the building facades, upper storeys in buildings, etc. as well as elements in public space, for example bridges, underpasses, elevations, platforms, ledges, etc. that influence our activity in such spaces. These prominent examples of the third dimension are almost insignificant, except for some few examples of skyscrapers or prominent topographical relief, when considering the city from a top-down perspective, as for example on a map. However, at the architectural scale, which focusses at the human scale, such features are what define space, and arguably affect the way space is used. Therefore, it was decided that an accurate simulation of PSA in this work would require the environments to be represented in their full three dimensions, and agents would indeed perceive their environment in 3D.

11.3 Contributions

This section will present a list of all contributions of this work, by offering a summary of each (in no particular order at the moment). Starting with secondary contributions first, this work presented:

A site surveying method for recording PSA in parks. The surveying method aims for efficiency, allowing a single surveyor using a field survey application on a smartphone to cover 100 hectares in approximately 90 minutes, capturing park visitor locations and the activities they are engaged in, split among 2 (and potentially more) activity types. Some alternatives were developed, discussing survey approaches de-

pending on terrain type and obscured visibility. A further GIS process was presented by which recorded activity was dispersed back into space.

An analysis of real-time datasets pertaining to activity in open public spaces on their own, as well as a correlation analysis between activity in parks (as captured in the aforementioned real-time datasets) and time and weather conditions. Through this analysis, it was found that: daily park visitor volumes follow a consistent day-night cycle as expected, and furthermore seem to be affected to a lesser degree by weather conditions such as cloud coverage and wind speed (an expected and logical outcome), but not temperature. Furthermore, it was established that real-time data needs to be available at large volume to perform any meaningful analysis.

A real-time forecast model of park visitor volume, using two approaches: a linear regression model using weather forecasts as the predicting variable and proxy real-time datasets of visitor activity as the predicted variable, and a naive forecast model. It was found that the naive model outperformed the Generalized Linear Model (GLM) in all cases, suggesting that real-time datasets relating to PSA, when examined at a fine temporal scale, exhibit a significant amount of variation/noise to be accurately predicted using a GLM.

A review of observational studies on human behaviour in public spaces. Surveys on the topic suggest an agreement on multiple aspects of overall human behaviour, concerning social characteristics (the majority of people in public spaces are found to be in groups of small numbers, with an average of approximately 1.7 people per group), locomotion (average movement speed was found to be 1.5-1.6 m/s negatively correlating with group size), movement (in open spaces, the direct path between current location and target location was found to be preferable), as well as crowding (people engaging in stationary activities in public spaces were found to prefer locations that placed them closer to others).

A review of the field of pedestrian ABMs in the past 15 years. The review highlighted trends in the field in recent years, identified mainly in increased fidelity

in agent perception and behaviours, and increased agent heterogeneity through a broadening of the field to incorporate other approaches. While no particular trend was identified in terms of spatial resolution, size, or dimensionality, models in recent years were found to include rules that enable agents to function with greater autonomy, namely by incorporating vision and a wider set of behavioural rules, therefore increasing fidelity from the bottom-up. An additional trend was noted in the turn towards other fields, incorporating psychological and personality traits to agents interacting in spatial environments, thus allowing for greater heterogeneity.

An ABM of PSA functioning in three-dimensional space, presented using the ODD framework (Grimm et al., 2010). This was achieved by extending existing approaches in pedestrian modelling and incorporating observations of human behaviour in public spaces as agent behavioural rules. This implementation constitutes a partial validation of observations and hypotheses on human behavioural rules, as presented in relevant literature.

The primary contribution of this work consists of a framework and general model for simulating activity in public spaces, in real-time. This *Real-Time Model of Public Space Activity* is the result of the combination of the three fields of this work: ABMs, PSU, and RTD. In developing such a model, this work highlighted recent advances in multiple fields including Real-Time Data and urban modelling, but also a rise in availability of 3D geometric data of cities, and even further a potential for the procedural generation of 3D city models at very high detail. This work identifies these advances in capturing and recreating the urban landscape not as an end-goal in urban visualisations, but rather as the stepping stone and basis for creating high-fidelity simulations of urban dynamics, as acted out by the city's inhabitants. This work therefore assumes that the various virtual 3D models of cities and places available through a plethora of platforms (Google Maps being the most well-known example, but also Mapbox, WRLD, Mantle, Vizicities, among others) will function as the virtual environment within which synthetic individuals may interact and recreate the daily urban experience. This work further assumes that RTD

on cities will continue to expand, first of all in volume and aspects captured, but also in terms of veracity and accuracy, powered by the Internet of Things (IoT) and ubiquitous sensing, and will therefore enable the aforementioned simulations to perform in real-time; to develop high-fidelity models and simulations of cities that run concurrently to the real world. This would enable then the transition from urban dashboards as monitors of urban routine, to urban simulations as predictors of an urban near-future.

11.4 Future Work

This work relied on multiple fields of study in its investigation of real-time models of activity in public spaces. It was expected then that any future paths this work may take can be identified in multiple fields. Furthermore, due to this work's combinatorial nature, it focussed on bringing multiple fields together. Therefore, it is acknowledged that there exists potential for improving this work just by exploring and incorporating advances in each of the fields of this work. However, even at this early stage for this *Real-Time Model of Public Space Activity*, potential applications have been identified. This section will address the future of this work, by discussing some of the areas this work can expand in, and by briefly presenting potential applications.

Regarding this work's computational nature as expressed mainly through the ABMs developed, the exploratory aspects of the models could certainly benefit by investigating scaling potential and capabilities. The models implemented in the two case studies focussed on well-defined areas covering a surface area of approximately 100 hectares, and were not tested in any aspect (neither model accuracy, nor computational capability) at capturing activity over larger areas. The first point of expansion of this work therefore is identifying the required *changes and optimizations needed to scale up the simulations*, in order to capture activity at the scale of the individual over a larger part of the urban environment. Improvements for such an endeavour are identified primarily in computational efficiency, which would require first of all

a review of algorithms with a view towards optimization, as well as the capabilities of other programming languages and modelling frameworks, more suited for large-scale simulations. However, in addition to the predominantly technical aspect of algorithm performance, scaling up the simulations presents another challenge that was not covered in this work, regarding land use. This work dealt exclusively with park activity for reasons discussed previously, and to do so it focussed on two of the largest open urban areas in London, Hyde Park and Queen Elizabeth Olympic Park, with parks themselves only covering a small percentage of the total area of London. Any expansion of the target area would inevitably encompass urbanized and built-up areas as well, which potentially present a largely different set of behaviours and rules in terms of user activity. Therefore, the second point of expansion of this work regarding scaling up the simulations is the *expansion of agent behavioural rules to include activity in urbanized/more complex areas*, such as plazas, squares, indoor spaces, and sidewalks. A third branch for this work is further identified in the combination of the two aims mentioned above with the increasing availability of 3D urban geometry, as found through multiple online mapping platforms. If an ABM is developed that can simulate varied public space activity over large areas with computational efficiency, then such a tool may be *coupled with procedural generation models of 3D urban geometry*, thereby producing simulations for potentially any location in urban space. This would require an abstraction of agent rules to enable automatic coupling and identification of relevant rule sets, and additional incorporation of procedural 3D environment generation, which was not covered in this work as the virtual environments for the areas of interest were recreated manually. Such a model would be able to leverage the potential of 3D mapping tools (e.g. openstreetmap, Mapbox), to produce expansive, detailed simulations of urban dynamics.

Regarding the Real-Time nature of this work, the overall conclusion was that publicly available real-time data sources at present do not yet offer the required fidelity to capture, forecast, and validate models of activity in public spaces continuously in real-time. With that said, this thesis acknowledges both its own limitation in mainly

focussing on publicly available datasets, as well as the rapidly expanding general field of Real-Time and Big Data. Therefore, this work will continue to *evaluate RTD sources regarding their potential in capturing activity in public spaces*, as they become available, as anticipates that such information will become available in the near-future, with the hope that it will be placed in the public domain. Furthermore, in the future, this work will re-evaluate the methods used in short-term forecasting and will perform a more detailed reading of available statistical methods and models, in order to *expand and enhance its forecasting arsenal* with tools in addition to the two approaches used here (naive forecasts and GLMs). Finally, as said previously, during this work, no dataset was found that accurately captured spatial activity, and therefore the potential of the SDM as a real-time tool remains un-validated. This may be revisited in the future, via two paths: First, by acquiring access to a dataset that provides such information, when it becomes available. Second, by developing appropriate methods *within the ABM framework* that will allow for real-time evaluation of spatial results and will enable a form of feedback in terms of agent spatial activity.

Moving forward from improvements, potential applications of this work will be discussed. Two main fields of application are identified here: one regarding the exploratory potential, related mainly to the spatial-computational nature of this work, the other focussing on dissemination. Regarding the exploratory potential, this work developed models of public space activity, calibrated at the level of the individual user of public space, and most importantly designed in a *reactive* ABM framework, meaning that agents react to their environment based on their codified set of behavioural rules. Such a model can be beneficial in the fields of urban design and planning, as a tool for exploring ”What If” scenarios, particularly in the design stage of urban public spaces. ABMs have been employed in these fields, and particularly pedestrian and crowd simulations have been used extensively in the design of large, crowded spaces such as airports, stadiums, and offices, to optimize flows, accessibility, and in evacuation scenarios. However, these models focus exclusively on individual movement, which as has been discussed previously is definitely not the

only (and potentially not even the primary) activity taking place in public spaces. As it stands, the design of new open urban spaces lacks a tool able to provide an evaluation of the design in terms of its stated goals, which often aim to be that of attracting people and activities, and allowing for comfortable and safe engaging in activities for all visitors. Guidelines on the design of public spaces exist and vary between cultures and locations, and are often aiming at allowing the individual designers to offer the best solution to the current problem. However, the evaluation of any implementation in urban design can often only be performed ”after the fact”, i.e. after design and construction is completed and the project is delivered to the public, which rules out any major corrections or improvements. A model of public space activity such as the one presented here could offer such a metric, covering basic functions required in successful public spaces, by allowing the designer to evaluate a proposed layout during the design stage, and help identify problematic issues. Of course, such an implementation would require extensive research to identify spatial activity qualities that are considered as ”good” to be used as a metric for design performance, which is in itself a complicated task, as preferences change over time, between cultures, and even between spaces in the same city (no two spaces of a city are exact duplicates, nor they should be). Nevertheless, a set of parameters may be identified that constitute a ”baseline” of performance in terms of public space activity, and then subsequently used as a metric to ensure that new urban spaces adequately address the basic needs of the community.

The second application regarding dissemination, is identified as a continuation and expansion upon existing approaches to visualisations of urban datasets. The volume and velocity of urban Big Data is being captured through *Urban Dashboards*, platforms offering a wide range of relevant information for a city at a glance. As has been discussed previously in this thesis, these dashboards perform well in visualising real-time information in a meaningful and comprehensible way to the general public, most often employing graph and chart visuals to disseminate information. Therefore, these platforms often lack the spatial aspect in their visualisations. On the other hand, online mapping platforms offer a view of the physical form of the

city *as it is*, i.e. a static image. These two approaches could potentially combine in spatial visualisations of the city *as it happens*, by appending real-time information to each location in the city. Indeed, some commercial mapping platforms have begun offering real-time information (e.g. Google Maps offers a live traffic view showing current traffic conditions, as well as current crowding conditions for venues, in metropolitan areas). It would be possible therefore to develop visualisation models spanning an entire city, able to visualise urban dynamics as they are exhibited through their individuals, in real-time, or in other words, function as a spatially-enhanced version of urban dashboards.

11.5 Concluding Remarks

Throughout this work, three main aspects of urban design and planning were placed in focus: how people interact with the urban environment (through the study of Public Space Activity (PSA) and observations on park visitor activity), tools for visualising and analyzing urban design (through the study of Agent-Based Models (ABMs) and 3D representations of urban public space), and data collection methods for studying urban activity (through the examination of urban Real-Time Data (RTD)). In addition to reviewing recent advances in these fields, this work presented methods for combining these fields, so that more comprehensive and detailed models of urban environments may be constructed, that can operate at finer spatial and temporal scales, and are capable of simulating aspects of the city *as it is right now*, in other words building a *'digital twin'* (Dawkins et al., 2018) of a city's public space. In this regard, this work presented an approach that builds on previous work on virtual 3D real-time models of cities, otherwise termed *'Urban Simulacra'* (Batty and Hudson-Smith, 2005) and *'Mirror Worlds'* (Hudson-Smith et al., 2009), proposing new ways for viewing, understanding, and planning future cities.

Part IV

Appendices

Appendix A

Auxiliary Functions

A.1 Disaggregating varying length time series

This paper discusses a linear interpolation method for disaggregating time series of varying duration. For a review of previous work on statistical disaggregation methods, see (Guerrero, 1990). Primarily, this method functions on a similar premise to that of previous work (Lisman and Sandee, 1964, Boot et al., 1967) using neighbouring periods to calculate disaggregated values for the period in question, and focuses on removing artificial steps potentially introduced between periods. Furthermore, it addresses the issue of working with varying period durations, where period lengths are known to be different. By employing a linear interpolation approach, absolute length is irrelevant, instead using relative positions in the period. As such, this method can be applied to time series consisting of different length periods, as it is in this case, for calendar months with durations between 28-31 days. Furthermore, this method is fairly straightforward in application, as it requires the calculation of 3 values for each period, allowing for quick implementation.

Assume a time series $T = (T_a, T_b, ...T_n)$, consisting of consecutive periods of varying lengths (durations) $L = (L_a, L_b, ...L_n)$, each with an associated value $P = (P_a, P_b, ...P_n)$. The average value for each period n is $V_n = P_n/L_n$ (Figure A.1).

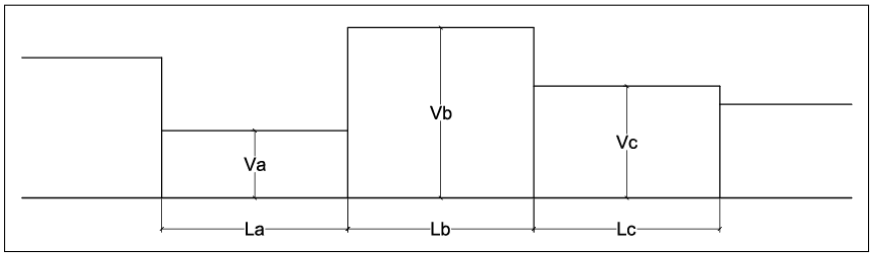


Figure A.1: Time Series with Period Totals

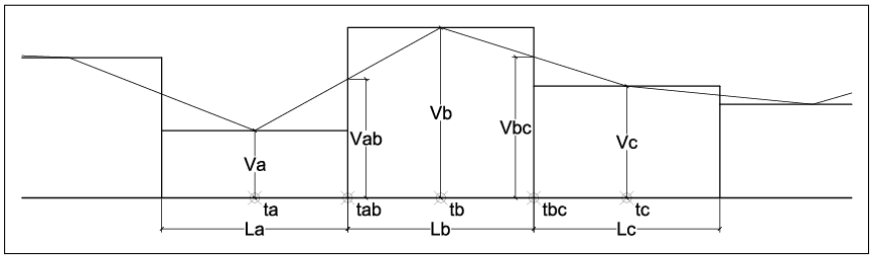


Figure A.2: Period Mid- and Break-points

For each period T_n, its midpoint in length is denoted by t_n. Breakpoints between periods are located at $t_{n,n+1} = t_n + (L_n/2)$. The values $V_{n,n+1}$ at breakpoints are calculated as a weighted average between periods T_n and T_{n+1} so that $V_{n,n+1} = V_n + (V_{n+1} - V_n) * \frac{t_{n,n+1} - t_n}{t_{n+1} - t_n}$ (Figure A.2).

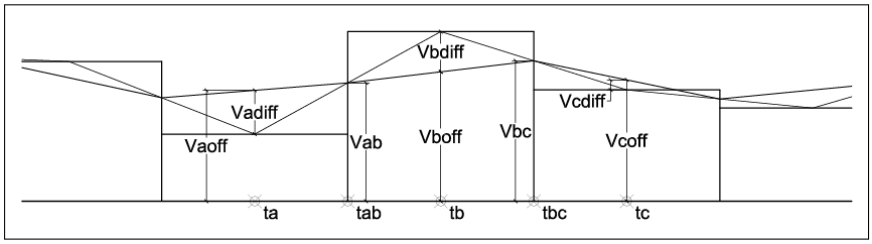


Figure A.3: Value Offsets

New values at midpoints t_n are calculated as averages between break points for a specific period, so that $V_{n.off} = \frac{V_{n-1,n} + V_{n,n+1}}{2}$. Furthermore, the difference $V_{n.diff}$ between period average and new value is calculated as $V_{n.diff} = V_n - V_{n.off}$ (Figure A.3).

The final value V'_n is calculated as the original value V_n offset by the difference $V_{n.diff}$, so that $V'_n = V_n + V_{n.diff}$, resulting in either positive or negative offset, depending on $V_{n.diff}$ value being positive or negative. The resulting series of vec-

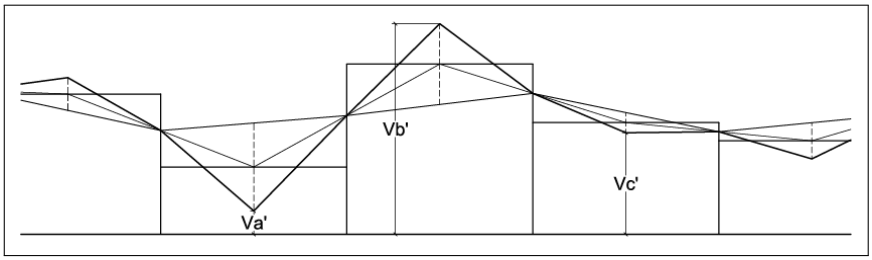


Figure A.4: Final Values

tors $(t_n, V'_n), (t_{n,n+1}, V'_{n,n+1})$..., produce a curve representing the disaggregated values (Figure A.4).

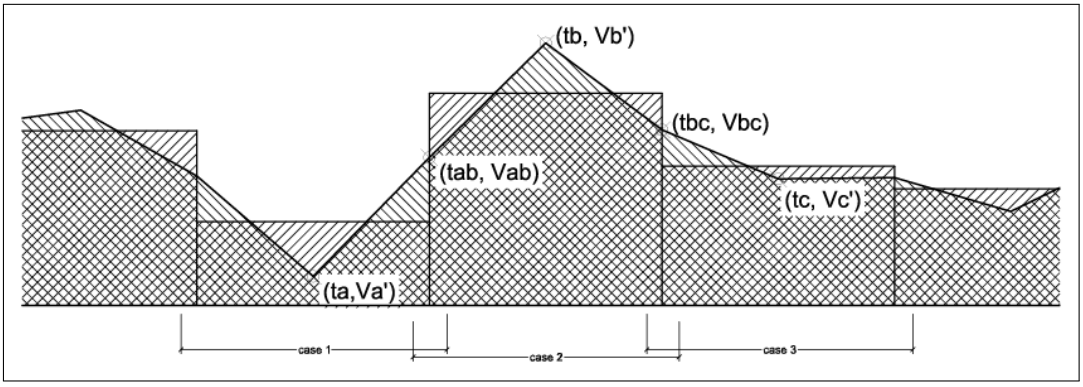


Figure A.5: Final Curve

To verify that sums are conserved for each period, the area defined by the curve must equal the area for the original shape (Figure A.5). Therefore, for case 2 (Figure A.6), it is enough to show that for each period, area $E_{aefd} = E_{abcd}$.

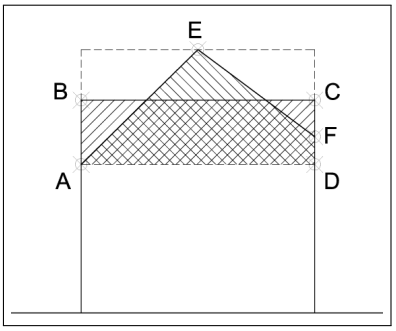


Figure A.6: Case 2 Detail

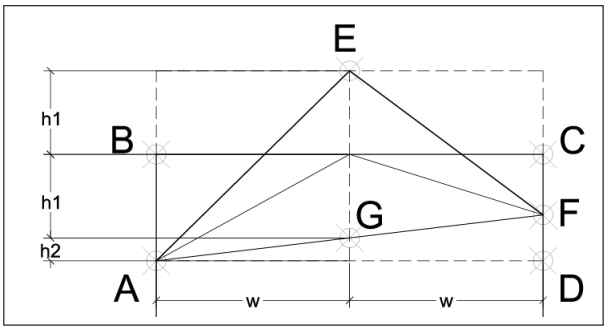


Figure A.7: Case 2 Area Equality

From Figure A.7:

$E_{abcd} = 2 * w * (h_1 + h_2)$

$E_{aefd} = E_{afd} + E_{aeg} + E_{egf}$

$E_{afd} = 2 * w * h_2$

$E_{aeg} = w * h_1$

$E_{egf} = w * h_1$

therefore:

$E_{aefd} = 2 * w * h_2 + 2 * w * h_1 = 2 * w * (h_1 + h_2)$

Similarly for cases 1 and 3, areas of resulting shapes equal original shapes. Therefore, to calculate the value V_x at time t_x in period T_n:

if $t_x < t_n$ (time point is in the first half of the period), then the value is a weighted average

$$V_x = V_{n-1,n} + (V_n - V_{n-1,n}) * \frac{t_x - t_{n-1,n}}{t_n - t_{n-1,n}}$$

Similarly if $t_x > t_n$, then

$$V_x = V_n + (V_{n,n+1} - V_n) * \frac{t_x - t_n}{t_{n,n+1} - t_n}$$

A.2 Point in Polygon Python Function

Code snippet illustrating the *Point in Polygon* function developed to be used within the Python environment. Given a point as a pair of coordinates `x, y`, and a list of coordinate pairs `poly`, the function returns `True` if the point is inside the polygon, and `False` if not. The code was originally published by John Berry on stackoverflow[1].

```
1 def point_in_poly(x,y,poly):
2
3     n = len(poly)
4     inside = False
5
6     p1x,p1y = poly[0]
7     for i in range(n+1):
```

[1] http://stackoverflow.com/questions/16325720/point-in-convex-polygon

```
        p2x,p2y = poly[i % n]
        if y > min(p1y,p2y):
            if y <= max(p1y,p2y):
                if x <= max(p1x,p2x):
                    if p1y != p2y:
                        xints = (y-p1y)*(p2x-p1x)/(p2y-p1y)+p1x
                    if p1x == p2x or x <= xints:
                        inside = not inside
        p1x,p1y = p2x,p2y

    return inside
```

An example list of coordinates outlining Hyde Park in London, listed in clockwise order:

```
[(51.502188500000003,-0.174641000000000),
(51.503152900000003,-0.174667300000000),
(51.506781400000001,-0.172041400000000),
(51.506699699999999,-0.171174900000000),
(51.507238999999998,-0.170492200000000),
(51.510082799999999,-0.170807300000000),
(51.511308499999998,-0.173170600000000),
(51.511962099999998,-0.173328100000000),
(51.513432700000003,-0.158675700000000),
(51.510327900000000,-0.156601300000000),
(51.505539200000001,-0.150981900000000),
(51.503381599999997,-0.150876900000000),
(51.503234499999998,-0.153424000000000),
(51.502188500000003,-0.166159500000000)]
```

A.3 Automated Social Media Data Collection

Social Media post collection was performed using automated scripts written in the python programming language. The scripts were set to run every day 15 minutes past midnight, and queried social media services' Application Programming Interfaces (APIs) for geolocated posts originating in the area of interest, published any time during the previous day. In the following script, data collection for Case Study 1: Hyde Park is shown, collection for Case Study 2: Queen Elizabeth Olympic Park used the same functions, with the only difference being the coordinate pairs. Script setup and global variables are set as following:

```
import requests
import threading
import json
import sys
from datetime import datetime
import time
import os
import tweepy

lat = 51.505770
```

```
lng = -0.164339

#Twitter authorisation setup
consumer_token = 'app consumer key goes here'
consumer_secret = 'app consumer secret goes here'
key = 'access token key goes here'
secret = 'access token secret goes here'

#Instagram authorisation setup
igAccessToken = 'instagram access token goes here'

maxTimeStart = int(time.time()) - 15*60
minTimeStart = maxTimeStart - 86400

maxTime = maxTimeStart
minTime = minTimeStart

events = []
counter = 0

looping = True

dateNow = time.strftime("%Y_%m_%d", time.localtime(maxTimeStart))

filename = "hyp_24H-" + dateNow
log = 'log.csv'

hypCorners = [(51.502188500000003,-0.174641000000000),
      (51.503152900000003,-0.174667300000000),
      (51.506781400000001,-0.172041400000000),
      (51.506699699999999,-0.171174900000000),
      (51.507238999999998,-0.170492200000000),
      (51.510082799999999,-0.170807300000000),
      (51.511308499999998,-0.173170600000000),
      (51.511962099999998,-0.173328100000000),
      (51.513432700000003,-0.158675700000000),
      (51.510327900000000,-0.156601300000000),
      (51.505539200000001,-0.150981900000000),
      (51.503381599999997,-0.150876900000000),
      (51.503234499999998,-0.153424000000000),
      (51.502188500000003,-0.166159500000000)]
```

A set of auxiliary functions was programmed, to setup individual platform collector authorisation, for performing basic spatial queries (point in polygon analysis), for processing individual tweet objects, sorting the daily list of posts chronologically, and finally writing to an external file. These auxiliary functions are as follows:

```
def loopSetup():
  global minTime, maxTime, events, counter, url, api, query, gCode
      , url1, url2, url3, url4, url5, loopStartTime
  loopStartTime = maxTime
  events = []
  counter = 0

  fg = open(log, 'w')
  fg.write(dateNow + '_started')
  fg.write('\n')
  fg.close()

  #TWEEPY LOOP SETUP
  auth = tweepy.OAuthHandler(consumer_token, consumer_secret)
  auth.set_access_token(key, secret)
  api = tweepy.API(auth)
```

```
  query = ''
  gCode = str(lat) + ',' + str(lng) + ',1.5km'

  #IG LOOP SETUP
  url1 = "https://api.instagram.com/v1/media/search?access_token
     ={0}&lat=".format(igAccessToken)
  url2 = "&lng="
  url3 = "&max_timestamp="
  url4 = "&min_timestamp="
  url5 = "&distance=1500&count=50"

  maxTimeLog = maxTime
  url = url1 + str(lat) + url2 + str(lng) + url3 + str(maxTime) +
     url4 + str(minTime) + url5

def point_in_poly(x,y,poly):
    n = len(poly)
    inside = False

    p1x,p1y = poly[0]
    for i in range(n+1):
        p2x,p2y = poly[i % n]
        if y > min(p1y,p2y):
            if y <= max(p1y,p2y):
                if x <= max(p1x,p2x):
                    if p1y != p2y:
                        xints = (y-p1y)*(p2x-p1x)/(p2y-p1y)+p1x
                    if p1x == p2x or x <= xints:
                        inside = not inside
        p1x,p1y = p2x,p2y
    return inside

def process_status(t):
  #process individual statuses
  d = t.created_at
  dts = int((d - datetime(1970,1,1)).total_seconds())
  return dts, t.created_at, t.coordinates, t.id

def sortList():
  global events, counter, minTimeStart, maxTimeStart
  events.append([])
  events[counter].append(00000)
  events[counter].append(minTimeStart)
  events[counter].append(lat)
  events[counter].append(lng)
  events[counter].append(-1)
  counter += 1

  events.append([])
  events[counter].append(11111)
  events[counter].append(maxTimeStart)
  events[counter].append(lat)
  events[counter].append(lng)
  events[counter].append(-1)
  counter += 1

  latMin = min(events, key=lambda events: events[2])[2]
  latMax = max(events, key=lambda events: events[2])[2]
  lonMin = min(events, key=lambda events: events[3])[3]
  lonMax = max(events, key=lambda events: events[3])[3]

  events = sorted(events, key=lambda events: events[1])

def writeFile():
  global filename, events

  length = len(events) - 2
```

```
81
82    os.chdir("data")
83    fgStr = filename + "-" + str(length) + '.csv'
84
85    fg = open(fgStr, 'w')
86    fg.write('uid,ts,lat,lon,src')
87
88    for event in events:
89      s = str(event[0]) +","+ str(event[1]) +","+ str(event[2])
           +","+ str(event[3]) +","+ str(event[4])
90      fg.write('\n')
91      fg.write(s)
92    fg.close()
93
94    with open(fgStr) as f:
95      with open("hyp_24H-latest.csv", "w") as f1:
96        for line in f:
97          f1.write(line)
98    os.chdir("..")
```

The following function collected all geolocated tweets returned by the Twitter API, that originated from within the area of interest in the last 24 hours from when the function was executed:

```
1  def collectorTweets():
2    global counter, api, query
3
4    twcounter = 0
5    validTweets = 0
6    endDay = False
7
8    searched_tweets = tweepy.Cursor(api.search, q=query, geocode=
        gCode, count=100).pages()
9
10   for page in searched_tweets:
11     for tweet in page:
12       twcounter += 1
13       procTweet = process_status(tweet)
14       if procTweet[0] - minTime > 0:
15         if procTweet[0] - maxTimeStart < 0:
16           if procTweet[2]:
17             uid = procTweet[3]
18             ts = procTweet[0]
19             lt = procTweet[2]["coordinates"][1]
20             lon = procTweet[2]["coordinates"][0]
21             lnk = "noLnk"
22
23             if point_in_poly(lt, lon, hypCorners):
24               events.append([])
25               events[counter].append(uid)
26               events[counter].append(ts)
27               events[counter].append(lt)
28               events[counter].append(lon)
29               events[counter].append(1)
30               counter += 1
31               validTweets += 1
32
33       else:
34         endDay = True
35         break
36     if endDay:
37       break
38
39     fg = open(log, 'a')
```

```
    fg.write(str(time.time()) + '_tw iter done')
    fg.write('\n')
    fg.close()
    time.sleep(65)
```

The following function was used to collect all geolocated instagram posts returned by the Instagram API, that originated from within the area of interest in the last 24 hours from when the function was executed:

```
def collectorInstagrams():
  global counter
  global maxTime, minTime, url, locIdCounter, locs, locsCounter

  url = url1 + str(lat) + url2 + str(lng) + url3 + str(maxTime) +
      url4 + str(minTime) + url5

  locIdCounter = 0
  locs = []
  locsCounter = []

  while maxTime > (minTime + 3600):
    response = requests.get(url)

    d = json.loads(response.text)
    data = d["data"]

    for i in range(0,len(data)-1):
      lt = data[i]["location"]["latitude"]
      lon = data[i]["location"]["longitude"]
      uid = data[i]["user"]["id"]
      ts = int(data[i]["created_time"])
      lnk = data[i]["link"]
      val = 1
      s = str(uid) +","+ str(ts) +","+ str(lt) +","+ str(lon)
          +","+ str(lnk) +","+ str(val)

      if "id" in data[i]["location"]:
        locIdCounter += 1
        locId = data[i]["location"]["id"]
        locNm = data[i]["location"]["name"]
        if locId in locs:
          locsCounter[locs.index(locId)] += 1
        else:
          locs.append(locId)
          locsCounter.append(1)

        if point_in_poly(lt, lon, hypCorners):
          events.append([])
          events[counter].append(uid)
          events[counter].append(ts)
          events[counter].append(lt)
          events[counter].append(lon)
          events[counter].append(0)
          counter += 1
      else:
        if point_in_poly(lt, lon, hypCorners):
          events.append([])
          events[counter].append(uid)
          events[counter].append(ts)
          events[counter].append(lt)
          events[counter].append(lon)
          events[counter].append(2)
          counter += 1
```

```
53
54        #end if no polygon yet
55
56      sg = ''
57      maxTime = int(data[len(data)-1]["created_time"])
58      url = url1 + str(lat) + url2 + str(lng) + url3 + str(maxTime)
           + url4 + str(minTime) + url5
59      # print('ig iteration done')
60      fg = open(log, 'a')
61      fg.write(str(time.time()) + '_ig iter done')
62      fg.write('\n')
63      fg.close()
64      time.sleep(2)
```

The following lines of code called the main functions in order and output the daily list to an external file.

```
1  def loop():
2    global loopStartTime, loopFunction, looping
3    print('___start')
4    loopSetup()
5    print('___start Tw')
6    collectorTweets()
7    print('___start Ig')
8    collectorInstagrams()
9    print('___start Sort')
10   sortList()
11   print('___start write')
12   writeFile()
13
14   fg = open(log, 'a')
15   fg.write(dateNow + '_finished')
16   fg.write('\n')
17   fg.close()
```

The following code was used to collect planned events and number of attendees, using Facebook's Graph Api. Graph API queries require search keywords to be provided, for the first case study the string 'hyde park' was used. Essentially this returned any events mentioning hyde park in any field. Further query parameters were used to filter the results, so that the potential hit should be of type *event* as set in Facebook's ecosystem, its stated location falling within within 2 kilometers of the centre of Hyde Park, and it starting within a specific time period. Further spatial filters were added to ensure the event was taking place within the park.

```
1  import requests
2  import facebook
3  import json
4  import csv
5  import sys
6  from datetime import datetime
7  import time
8
9  clientID = "clientID"
10 clientSecret = "clientSecret"
```

```
accessToken = "accessToken"
userAccessToken = 'userAccessToken'
longLifeUserAccessToken = 'longLifeUserAccessToken'

hypCorners = [(51.502188500000003,-0.174641000000000),
      (51.503152900000003,-0.174667300000000),
      (51.506781400000001,-0.172041400000000),
      (51.506699699999999,-0.171174900000000),
      (51.507238999999998,-0.170492200000000),
      (51.510082799999999,-0.170807300000000),
      (51.511308499999998,-0.173170600000000),
      (51.511962099999998,-0.173328100000000),
      (51.513432700000003,-0.158675700000000),
      (51.510327900000000,-0.156601300000000),
      (51.505539200000001,-0.150981900000000),
      (51.503381599999997,-0.150876900000000),
      (51.503234499999998,-0.153424000000000),
      (51.502188500000003,-0.166159500000000)]


timeNow = int(time.time())
queryStartTime = 1447027201
queryEndTime = 1447804799

queryDurationSecs = queryEndTime - queryStartTime + 10
queryDurationDays = round(queryDurationSecs / 86400)

def point_in_poly(x,y,poly):
    n = len(poly)
    inside = False

    p1x,p1y = poly[0]
    for i in range(n+1):
        p2x,p2y = poly[i % n]
        if y > min(p1y,p2y):
            if y <= max(p1y,p2y):
                if x <= max(p1x,p2x):
                    if p1y != p2y:
                        xints = (y-p1y)*(p2x-p1x)/(p2y-p1y)+p1x
                    if p1x == p2x or x <= xints:
                        inside = not inside
        p1x,p1y = p2x,p2y
    return inside

graph = facebook.GraphAPI(access_token=longLifeUserAccessToken,
    version = '2.5')
r = graph.request('search', args = {'q': 'hyde park',
  'type': 'event',
  'center':'51.505770,-0.164339',
  'distance':'2000',
  'since': queryStartTime,
  'until': queryEndTime,
  'limit': '500'})

cT = 0
cLoc = 0
cHyP = 0

events = []
for d in r['data']:
  cT += 1

  try:
    lat = d['place']['location']['latitude']
    lon = d['place']['location']['longitude']
    cLoc += 1
  except:
    continue
```

```
78
79    if point_in_poly(lat, lon, hypCorners):
80      atts = 0
81      req = d['id'] + '/attending/'
82
83      try:
84        print(req,
85          d['start_time'],
86          d['end_time']
87          )
88      except:
89        print(req,
90          d['start_time'],
91          )
92
93      eventDatetime = datetime.strptime(d['start_time'], '%Y-%m-%dT%
            H:%M:%S%z')
94      eventDatetime = eventDatetime.replace(tzinfo=None)
95      ts = int((eventDatetime - datetime(1970,1,1)).total_seconds())
96      print(ts)
97      e = graph.request(req, args = {"limit": '500'})
98      while(True):
99        try:
100         atts += len(e['data'])
101         # Attempt to make a request to the next page of data, if
              it exists.
102         e=requests.get(e['paging']['next']).json()
103       except KeyError:
104         # When there are no more pages (['paging']['next']), break
                from the
105         # loop and end the script.
106         break
107
108     events.append([])
109     events[cHyP].append(d['id'])
110     events[cHyP].append(ts)
111     events[cHyP].append(lat)
112     events[cHyP].append(lon)
113     events[cHyP].append(atts)
114     cHyP += 1
115
116 datesEvents = []
117 t = queryStartTime
118 for i in range(0,queryDurationDays):
119   datesEvents.append([])
120   datesEvents[i].append(t)
121   datesEvents[i].append(0)
122   datesEvents[i].append(0)
123   for e in events:
124     if e[1] > t:
125       if e[1] < t + 86400:
126         datesEvents[i][1] += 1
127         datesEvents[i][2] += e[4]
128   t += 86400
129 print(len(datesEvents))
130
131 filename = 'fb-hyp-events-socmWthrRange_' + str(timeNow) + '.csv'
132 with open(filename, 'w', newline='') as testfile2:
133   csv_writer = csv.writer(testfile2)
134   csv_writer.writerows(datesEvents)
```

A.4 Weather Conditions Data Collection

Information on weather conditions at an area of interest was collected using the web API service 'forecast.io', via an automated script written in the python programming language. A python library was used (*forecastio*) as an interface, and collected data was subsequently stored in a JSON file. The full code used is as follows:

```
import datetime
import forecastio
import json

api_key = "apiKey"
lat = 51.505770
lng = -0.164339

d = datetime.datetime.utcnow()
epoch = datetime.datetime.utcfromtimestamp(0)
s = int((d - epoch).total_seconds()) - 3600
d = datetime.datetime.utcfromtimestamp(s - 43200)
dateNow = d.strftime("%Y_%m_%d")
filename = "hyp_24H-" + dateNow + "-weather"

def weatherCollection(d):
  forecast = forecastio.load_forecast(api_key, lat, lng, time = d)

  with open(filename + ".json", 'w') as outfile:
    json.dump(forecast.json, outfile)

  with open("hyp_24H-latest-weather.json", 'w') as outfile:
    json.dump(forecast.json, outfile)

weatherCollection(d)
```

A.5 QGIS Python Functions

This section presents code written in the Python programming language (version 2.7.5), for use in the QGIS software (latest tested version: 2.18.16), in order to extend and include additional functionality, not readily available through the core tool set of QGIS.

A.5.1 Tree Planting Script

The following python code was used to add tree locations for areas marked as 'wood' in OpenStreetMap. Tree densities were calculated from other areas where tree point locations were available.

```
import random,sys
import math

layerWoods = QgsMapLayerRegistry.instance().mapLayersByName("
    hypOsm2-features-woodFinal-merc")[0]
layerTrees = QgsMapLayerRegistry.instance().mapLayersByName("
    hypOsm2-trees-merc")[0]

def euclideanDistance(point1,point2):
    return math.sqrt((point2.x()-point1.x())**2 + (point2.y()-
        point1.y())**2)

def randomMoveWithinRadius(p, r):
    xNew = p[0] + (random.random()-0.5) * 2 * r
    yNew = p[1] + (random.random()-0.5) * 2 * r
    pNew = QgsPoint(xNew,yNew)
    while(euclideanDistance(p,pNew) > r):
        xNew = p[0] + (random.random()-0.5) * 2 * r
        yNew = p[1] + (random.random()-0.5) * 2 * r
        pNew = QgsPoint(xNew,yNew)
    return pNew

def randomMoveWithinRadiusNormalDist(p, r):
    xNew = p[0] +random.normalvariate(0,0.333) * r
    yNew = p[1] + random.normalvariate(0,0.333) * r
    pNew = QgsPoint(xNew,yNew)
    d = euclideanDistance(p,pNew)
    print(d)
    while(d > r):
        xNew = p[0] + random.normalvariate(0,0.333) * r
        yNew = p[1] + random.normalvariate(0,0.333) * r
        pNew = QgsPoint(xNew,yNew)
        d = euclideanDistance(p,pNew)
    return pNew

def randomPointInBoundingBox(b):
    xNew = random.uniform(b.xMinimum(), b.xMaximum())
    yNew = random.uniform(b.yMinimum(), b.yMaximum())
    pNew = QgsPoint(xNew,yNew)
    return pNew

def pointIsInArea(point):
    if(selectedArea.geometry().contains(point)):
        return True
    else:
        return False

def pointIsInFeature(point, feature):
    return feature.geometry().contains(point)

layer = layerWoods
features = layer.selectedFeatures()

if layerTrees.isEditable():
    idx = layer.fieldNameIndex('treesInt')
    for f in features:
        treeAmt = f.attributes()[idx]

        geom = f.geometry()
        bb = geom.boundingBox()
        for i in range(0,treeAmt):

            p = randomPointInBoundingBox(bb)
            inArea = pointIsInFeature(p, f)
            while (inArea == False):
                p = randomPointInBoundingBox(bb)
                inArea = pointIsInFeature(p, f)
```

```
65
66                fNew = QgsFeature()
67                fNew.setAttributes([0, random.randrange(0,5000)])
68                fNew.setGeometry(QgsGeometry.fromPoint(p))
69
70                if(layerTrees.isEditable()):
71                    (res, outFeats) = layerTrees.dataProvider().
                          addFeatures([fNew])
72                else:
73                    print("layer not editable")
74        iface.mapCanvas().refresh()
75    else:
76        print("layer not editable")
77
78    layerTrees.commitChanges()
```

The overall reconstruction process along with the final result of all tree locations used in the model is seen in Figure A.8.

(a) Tree Point Locations and areas marked 'wood' **(b)** Reconstructed Tree Locations

Figure A.8: Tree Reconstruction Process

A.5.2 ABM Validation Grid Script

The following code was used to create the validation grids used for the *Expanding Cell Validation Method* in Chapters 8 and 9. It is set up in such a way to generate grids at multiple scales, and performs the cell counts and error calculations automatically.

```
1  lSim = QgsMapLayerRegistry.instance().mapLayersByName('
       simulationResultsLayerName')[0]
2  lObs = QgsMapLayerRegistry.instance().mapLayersByName('
       observationsLayerName')[0]
3  se = lSim.extent()
4  so = lObs.extent()
5
6  xMin = min(se.xMinimum(), so.xMinimum())
7  xMax = max(se.xMaximum(), so.xMaximum())
```

```
yMin = min(se.yMinimum(), so.yMinimum())
yMax = max(se.yMaximum(), so.yMaximum())

xRange = xMax - xMin
yRange = yMax - yMin

def cellGenerator(gridxsize, gridysize, x, y, xShift, yShift):
    if (xShift < 0):
        shiftxmin = xMin - gridxsize * 0.25
    elif(xShift > 0):
        shiftxmin = xMin - gridxsize * 0.75
    else:
        shiftxmin = xMin

    if (yShift < 0):
        shiftymin = yMin - gridysize * 0.25
    elif(yShift > 0):
        shiftymin = yMin - gridysize * 0.75
    else:
        shiftymin = yMin

    bl = QgsPoint(shiftxmin + x*gridxsize, shiftymin + y*gridysize
        )
    tl = QgsPoint(shiftxmin + x*gridxsize, shiftymin + y*gridysize
         + gridysize)
    tr = QgsPoint(shiftxmin + x*gridxsize + gridxsize, shiftymin +
         y*gridysize + gridysize)
    br = QgsPoint(shiftxmin + x*gridxsize + gridxsize, shiftymin +
         y*gridysize)
    cellGeom = QgsGeometry.fromPolygon([[bl,tl,tr,br]])
    fet = QgsFeature()
    fet.setGeometry(cellGeom)
    return fet


def createLayer(subdivs):
    vl = QgsVectorLayer("Polygon?crs=epsg:32630", "ValidationGrid-
        dayType_{0}".format(str(subdivs)), "memory")

    pr = vl.dataProvider()
    vl.startEditing()

    pr.addAttributes( [ QgsField("xShift",  QVariant.Int),
                    QgsField("yShift", QVariant.Int),
                    QgsField("countObs", QVariant.Int),
                    QgsField("countSim", QVariant.Int),
                    QgsField("pctObs", QVariant.Double),
                    QgsField("pctSim", QVariant.Double),
                    QgsField("pctDiff", QVariant.Double),
                    QgsField("pctDiffAbs", QVariant.Double),
                    QgsField("width_m", QVariant.Double),
                    QgsField("length_m", QVariant.Double),
                    QgsField("area_hec", QVariant.Double)
                    ] )

    gridxsize = xRange / subdivs
    gridysize = yRange / subdivs

    xShift = 0
    yShift = 0
    for xShift in [-1,1]:
        for x in range(subdivs+1):
            for y in range(subdivs):
                fet = cellGenerator(gridxsize, gridysize, x, y,
                    xShift, yShift)
```

```
67                  fet.setAttributes( [xShift,yShift
                        ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] )
68                  pr.addFeatures( [ fet ] )
69
70      xShift = 0
71      yShift = 0
72      for yShift in [-1,1]:
73          for x in range(subdivs):
74              for y in range(subdivs+1):
75                  fet = cellGenerator(gridxsize, gridysize, x, y,
                        xShift, yShift)
76                  fet.setAttributes( [xShift,yShift
                        ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] )
77                  pr.addFeatures( [ fet ] )
78
79      xShift = 0
80      yShift = 0
81      for x in range(subdivs):
82          for y in range(subdivs):
83              fet = cellGenerator(gridxsize, gridysize, x, y, xShift
                    , yShift)
84              fet.setAttributes( [xShift,yShift
                    ,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] )
85              pr.addFeatures( [ fet ] )
86
87      vl.commitChanges()
88      QgsMapLayerRegistry.instance().addMapLayer(vl)
89      return vl
90
91  def countPoints(lGrid):
92      lGrid.startEditing()
93      lGrid.updateFields()
94
95      totalObs = lObs.featureCount()
96      totalSim = lSim.featureCount()
97      obs = lObs.getFeatures()
98      sim = lSim.getFeatures()
99
100     for f in lGrid.getFeatures():
101
102         countObs = 0
103         countSim = 0
104         pctObs = 0.0
105         pctSim = 0.0
106         pctDiff = 0.0
107         pctDiffAbs = 0.0
108
109         obs = lObs.getFeatures()
110         sim = lSim.getFeatures()
111
112         fg = f.geometry()
113         for fo in obs:
114             if (fg.contains(fo.geometry())):
115                 countObs += 1
116
117         for fs in sim:
118             if (fg.contains(fs.geometry())):
119                 countSim += 1
120
121         pctObs = float(countObs) / (totalObs * 1.0)
122         pctSim = float(countSim) / (totalSim * 1.0)
123         pctDiff = pctObs - pctSim
124         pctDiffAbs = abs(pctDiff)
125
126         geom = fg
127         h = geom.boundingBox().height()
128         w = geom.boundingBox().width()
```

```
129         a = geom.area()/10000
130
131         f['countObs'] = countObs
132         f['countSim'] = countSim
133         f['pctObs'] = pctObs
134         f['pctSim'] = pctSim
135         f['pctDiff'] = pctDiff
136         f['pctDiffAbs'] = pctDiffAbs
137         f['width_m'] = w
138         f['length_m'] = h
139         f['area_hec'] = a
140         lGrid.updateFeature(f)
141
142     lGrid.commitChanges()
143
144 for subdivs in range(1,16):
145     l = createLayer(subdivs)
146     countPoints(l)
147     print('DONE WITH {0}'.format(l.name()))
```

A.5.3 Survey Activity Re-Dispersion Script

The following code was used to re-disperse visitor locations from the surveyor path to the surrounding area. This was needed as the locations of visitors captured during the site surveys was recorded as being on the surveyor path, rather than their actual location, as discussed in Section 6.3. The following is the second amended version of the script which was applied to Case Study 2: Queen Elizabeth Olympic Park (CS2:QEOP), which had the added requirement of a point being in the same survey area before and after transformation in addition to a distance limitation. The version applied to Case Study 1: Hyde Park (CS1:HyP) is identical with the only difference being the exclusion of the *pointIsInArea*() check.

```
1  import random,sys
2  import math
3
4  moveDist = 100
5  attempts = 10
6
7  layerWater  = QgsMapLayerRegistry.instance().mapLayersByName("
       waterFeaturesLayer")[0]
8  layerAreas = QgsMapLayerRegistry.instance().mapLayersByName("
       CS2QEOP-SurveyAreasLayer")[0]
9  selectedArea = layerAreas.selectedFeatures()[0]
10
11 def euclideanDistance(point1,point2):
12     return math.sqrt((point2.x()-point1.x())**2 + (point2.y()-
           point1.y())**2)
13
14 def randomMoveWithinRadius(p, r):
15     xNew = p[0] + (random.random()-0.5) * 2 * r
16     yNew = p[1] + (random.random()-0.5) * 2 * r
17     pNew = QgsPoint(xNew,yNew)
18     while(euclideanDistance(p,pNew) > r):
```

```
19          xNew = p[0] + (random.random()-0.5) * 2 * r
20          yNew = p[1] + (random.random()-0.5) * 2 * r
21          pNew = QgsPoint(xNew,yNew)
22      return pNew
23
24  def randomMoveWithinRadiusNormalDist(p, r):
25      xNew = p[0] +random.normalvariate(0,0.333) * r
26      yNew = p[1] + random.normalvariate(0,0.333) * r
27      pNew = QgsPoint(xNew,yNew)
28      d = euclideanDistance(p,pNew)
29      print(d)
30      while(d > r):
31          xNew = p[0] + random.normalvariate(0,0.333) * r
32          yNew = p[1] + random.normalvariate(0,0.333) * r
33          pNew = QgsPoint(xNew,yNew)
34          d = euclideanDistance(p,pNew)
35      return pNew
36
37  def pointIsInWater(point):
38      for f in layerWater.getFeatures():
39          if(f.geometry().contains(point)):
40              return True
41      return False
42
43  def pointIsInArea(point):
44      if(selectedArea.geometry().contains(point)):
45          return True
46      else:
47          return False
48
49  def featureIsGPS(feature):
50      return feature['event'] == 'GPS'
51
52  layer = iface.activeLayer()
53  features = layer.selectedFeatures()
54
55  if layer.isEditable():
56      for f in features:
57          fid = f.id()
58          if(featureIsGPS(f) == False):
59              geom = f.geometry()
60              xy = geom.asPoint()
61              print(xy, pointIsInArea(xy), pointIsInWater(xy),
                    featureIsGPS(f))
62              geomNew = QgsGeometry.fromPoint(
                    randomMoveWithinRadiusNormalDist(xy, moveDist))
63              inArea = pointIsInArea(geomNew.asPoint())
64              inWater = pointIsInWater(geomNew.asPoint())
65
66              c = 0
67              while((inArea==False or inWater==True) and c <
                    attempts):
68                  geomNew = QgsGeometry.fromPoint(
                        randomMoveWithinRadius(xy, moveDist))
69                  inArea = pointIsInArea(geomNew.asPoint())
70                  inWater = pointIsInWater(geomNew.asPoint())
71                  print(geomNew.asPoint(), inArea, inWater)
72                  c+=1
73              if (c < attempts):
74                  layer.changeGeometry(fid, geomNew)
75      iface.mapCanvas().refresh()
76  else:
77      print("Layer not in Edit Mode")
```

Appendix B

ABM Functions

This Appendix contains code written for the implementation of the Agent-Based Model (ABM) of Public Space Activity (PSA). The code was written using the C# programming language (.NET Version 2.0.50727.1433), and tested using Unity software (latest Unity version tested: Unity 5.6.2f1).

B.1 Agent Functions

The following function initializes an individual agent entity. The *AgentInit()* function is called by the simulation controller every time a new agent is introduced in the simulation.

```
1  public void AgentInit(float _radiusVisionMultiplier = 2f,
2      float _sitterChance = 0.15f,
3      float _featureVisitChance = 0.41f,
4      float _sportsChance = 0.05f,
5      float _radiusSports = 20f,
6      float _viewAngle = 90f)
7    {
8
9      radiusVisionMultiplier = _radiusVisionMultiplier;
10     sitterChance = _sitterChance;
11     featureVisitChance = _featureVisitChance;
12     sportsChance = _sportsChance;
13     radiusSports = _radiusSports;
14     viewAngle = _viewAngle;
15
16     if (randomizeSpeed)
17       speed += Random.Range (-0.4f, 0.4f);
18     if (!this.gameObject.GetComponent<UnityEngine.AI.NavMeshAgent>
           ())
19       this.gameObject.AddComponent<UnityEngine.AI.NavMeshAgent> ()
           ;
20     nma = this.GetComponent<UnityEngine.AI.NavMeshAgent> ();
```

```
21      nma.speed = speed;
22      nma.enabled = false;
23
24      lifetime = agentUtils.GetAgentLifetime ();
25      birthTime = Time.frameCount;
26
27      groupSize = agentUtils.GetGroupSize ();
28      for (int i = 0; i < groupSize; i++) {
29        this.gameObject.AddComponent<BoxCollider> ();
30      }
31
32      moveToRandomLocation ();
33      controller c = GameObject.FindGameObjectWithTag ("
           GameController").GetComponent<controller> ();
34      c.agentsWalking += groupSize;
35
36      hasInitiated = true;
37      Invoke ("DebugPath", Random.Range(0f,5f));
38    }
```

The *Update()* function is called once per frame in the Unity .NET environment. It controls agents' frame-by-frame behaviour, colours them according to current state, and keeps track of time spent on activities. It does not contain any decision trees. Unless an agent is engaged in a stationary activity, the *Update()* function calls the *TakeStep()* function, which moves an agent through the environment.

```
1  void Update () {
2    if (!hasInitiated) {
3      return;
4    }
5    timeAlive++;
6
7    if (agentState == simUtils.agentStates.Walking
8      || agentState == simUtils.agentStates.MovingToSittingSpot
9      || agentState == simUtils.agentStates.MovingToSportsSpot
10       || agentState == simUtils.agentStates.MovingToFeature
11       || agentState == simUtils.agentStates.
            SearchingNextSittingSpot
12     || agentState == simUtils.agentStates.SearchingNextSportsSpot
13       || agentState == simUtils.agentStates.Exiting)
14     TakeStep ();
15   else if (agentState == simUtils.agentStates.BuggedOut) {
16     decideNextActivity ();
17   }
18
19   if (agentState == simUtils.agentStates.Walking) {
20     GetComponent<Renderer> ().material.color = Color.white;
21   } else if (agentState == simUtils.agentStates.
        SearchingNextSittingSpot) {
22     GetComponent<Renderer> ().material.color = Color.white;
23     timeSpentPrepping++;
24   } else if (agentState == simUtils.agentStates.
        MovingToSittingSpot) {
25     GetComponent<Renderer> ().material.color = Color.white;
26     timeSpentPrepping++;
27   } else if (agentState == simUtils.agentStates.Sitting) {
28     GetComponent<Renderer> ().material.color = Color.blue;
29     timeSpentSitting++;
30   } else if (agentState == simUtils.agentStates.Sports) {
31     GetComponent<Renderer> ().material.color = Color.cyan;
```

```
32      timeSpentSportsing++;
33    } else if (agentState == simUtils.agentStates.FeatureVisitor) {
34      GetComponent<Renderer> ().material.color = Color.yellow;
35      timeSpentFeaturing++;
36    } else if (agentState == simUtils.agentStates.
          SearchingNextSportsSpot) {
37      timeSpentPrepping++;
38    } else if (agentState == simUtils.agentStates.MovingToSportsSpot
          ) {
39      timeSpentPrepping++;
40    } else if (agentState == simUtils.agentStates.MovingToFeature) {
41      timeSpentPrepping++;
42    }
43
44    timePctSpentSitting = timeSpentSitting / (float)timeAlive;
45    timePctSpentFeaturing = timeSpentFeaturing / (float)timeAlive;
46    timePctSpentSportsing = timeSpentSportsing / (float)timeAlive;
47    timePctSpentPrepping = timeSpentPrepping / (float)timeAlive;
48  }
```

```
1   void TakeStep(){
2     if (Vector3.Distance (this.transform.position, currentWalkTarget
          ) < 3f)
3       finishedMoving ();
4     else {
5       if (Vector3.Distance (this.transform.position, nextPathPoint)
            < 2f) {
6         try{
7         pathPoints.RemoveAt (0);
8         nextPathPoint = pathPoints [0];
9         }
10        catch{
11          timeAlive = lifetime;
12          finishedMoving ();
13        }
14      }
15
16      this.transform.LookAt (nextPathPoint);
17      this.transform.position += this.transform.forward * speed;
18    }
19  }
```

The agent randomly chooses a new activity every time it completes its current overall task. The new activity is chosen using a stochastic model based on a Probabilistic Finite-State Machine (PFSM), shown in Figure 7.6. Its code implementation is as follows:

```
1   void decideNextActivity(){
2     int timeLeft = lifetime - timeAlive;
3     int sitDuration = (int)avgActivityDuration*2;
4
5     if (timeAlive > lifetime) {
6       PrepareForExit ();
7       return;
8     } else if (timeAlive < 300) {
9       agentState = simUtils.agentStates.Walking;
10      moveToRandomLocation ();
11      return;
12    }
13
14    simUtils.agentActivities potentialNextActivity;
15    float v = Random.Range (0f, 1f);
```

```
16    if (v < sitterChance)
17      potentialNextActivity = simUtils.agentActivities.Sit;
18    else if (v < sitterChance + featureVisitChance && GameObject.
          FindGameObjectWithTag("GameController").GetComponent<
          controller>().featuresExist)
19      potentialNextActivity = simUtils.agentActivities.FeatureVisit;
20    else if (v < sitterChance + featureVisitChance + sportsChance)
21      potentialNextActivity = simUtils.agentActivities.Sports;
22    else
23      potentialNextActivity = simUtils.agentActivities.Walk;
24
25    nextActivity = potentialNextActivity;
26
27    if (nextActivity == simUtils.agentActivities.Sit) {
28      currentPrepStartTime = Time.frameCount;
29      agentState = simUtils.agentStates.SearchingNextSittingSpot;
30      currentActivity = simUtils.agentActivities.Walk;
31      nextActivityDuration = sitDuration;
32      StartCoroutine (SampleForSittingSpots ());
33    } else if (nextActivity == simUtils.agentActivities.FeatureVisit
          ) {
34      currentPrepStartTime = Time.frameCount;
35      currentActivity = simUtils.agentActivities.Walk;
36      nextActivityDuration = sitDuration;
37      setupFeatureVisit ();
38    } else if (nextActivity == simUtils.agentActivities.Sports) {
39      currentPrepStartTime = Time.frameCount;
40      agentState = simUtils.agentStates.SearchingNextSportsSpot;
41      currentActivity = simUtils.agentActivities.Walk;
42      nextActivityDuration = sitDuration;
43      StartCoroutine (SampleForSportsSpots ());
44    } else {
45      agentState = simUtils.agentStates.Walking;
46      moveToRandomLocation ();
47    }
48  }
```

Every time an agent completes an activity, the appropriate function is executed, either to set up the next set of directions (for example during a more complex process requiring preparation), or to return the agent to its default state of deciding its next activity. The two functions that take care of these are presented in the following code snippets (all stationary activities, including Sit, Feature Visit, and Sport, make use of the *finishedSitting()* function). The *finishedMoving()* function further illustrates the implementation for the calculation of the duration of a stationary activity.

```
1  void finishedMoving(){
2    if (Vector3.Distance (this.transform.position, currentWalkTarget
         ) > 10) {
3      setPathToLocation (currentWalkTarget);
4      return;
5    }
6    controller c = GameObject.FindGameObjectWithTag ("GameController
         ").GetComponent<controller> ();
7
8    if (agentState == simUtils.agentStates.Exiting)
9      RemoveAgent ();
10   else if (timeAlive > lifetime)
11     PrepareForExit ();
```

```
12    else if (agentState == simUtils.agentStates.
          SearchingNextSittingSpot) {
13      if (samplingLocations)
14        moveToRandomLocation ();
15      else {
16        agentState = simUtils.agentStates.MovingToSittingSpot;
17        setPathToLocation (nextSittingLocation);
18        locationToReturnTo = nextSittingLocation;
19      }
20    }
21    else if (agentState == simUtils.agentStates.MovingToSittingSpot)
          {
22      c.agentsSitting += groupSize;
23      c.agentsWalking -= groupSize;
24      agentState = simUtils.agentStates.Sitting;
25      currentActivity = simUtils.agentActivities.Sit;
26      nextActivity = simUtils.agentActivities.Walk;
27      nextActivityDuration = (int)(Time.frameCount -
            currentPrepStartTime);
28
29      float v1 = 1f / (sitterChance + sportsChance +
            featureVisitChance);
30      float v2 = (float) nextActivityDuration / avgActivityDuration;
31      float mod = ((v1 + v2 - 1) / (v1 - 1));
32      float durationFinal = (avgActivityDuration * mod * 1.5f);
33
34      StartCoroutine(InvokeAfterFrames ("finishedSitting", (int)
            durationFinal));
35    }
36    else if (agentState == simUtils.agentStates.
          SearchingNextSportsSpot) {
37      if (samplingSportsLocations)
38        moveToRandomLocation ();
39      else {
40        agentState = simUtils.agentStates.MovingToSportsSpot;
41        setPathToLocation (nextSportsLocation);
42        locationToReturnTo = nextSportsLocation;
43      }
44    }
45    else if (agentState == simUtils.agentStates.MovingToSportsSpot)
          {
46      c.agentsSports += groupSize;
47      c.agentsWalking -= groupSize;
48      agentState = simUtils.agentStates.Sports;
49      currentActivity = simUtils.agentActivities.Sports;
50      nextActivity = simUtils.agentActivities.Walk;
51      nextActivityDuration = (int)(Time.frameCount -
            currentPrepStartTime);
52      this.transform.localScale = new Vector3(radiusSports*2,10,
            radiusSports*2);
53
54      float v1 = 1f / (sitterChance + sportsChance +
            featureVisitChance);
55      float v2 = (float) nextActivityDuration / avgActivityDuration;
56      float mod = ((v1 + v2 - 1) / (v1 - 1));
57      float durationFinal = (avgActivityDuration * mod * 1.5f);
58
59      StartCoroutine(InvokeAfterFrames ("finishedSitting", (int)
            durationFinal));
60    }
61    else if (agentState == simUtils.agentStates.MovingToFeature) {
62      c.agentsFeatureVisitors += groupSize;
63      c.agentsWalking -= groupSize;
64      Vector3 fPos = targetFeature.transform.position;
65      Vector3 fExt = targetFeature.GetComponent<Collider> ().bounds.
            extents;
```

```
66      Vector3 newPos = new Vector3 (fPos.x + Random.Range(-fExt.x,
           fExt.x), fPos.y, fPos.z + Random.Range(-fExt.z,fExt.z));
67      int counter = 0;
68      while (counter < 30 && !simUtils.PointInOABB (newPos,
           targetFeature.GetComponent<BoxCollider> ())) {
69        counter++;
70        newPos = new Vector3 (fPos.x + Random.Range (-fExt.x, fExt.x
             ), fPos.y, fPos.z + Random.Range (-fExt.z, fExt.z));
71      }
72      this.transform.position = newPos;
73      agentState = simUtils.agentStates.FeatureVisitor;
74      currentActivity = simUtils.agentActivities.FeatureVisit;
75      nextActivity = simUtils.agentActivities.Walk;
76      nextActivityDuration = (int)(Time.frameCount -
           currentPrepStartTime);
77
78      float v1 = 1f / (sitterChance + sportsChance +
           featureVisitChance);
79      float v2 = (float) nextActivityDuration / avgActivityDuration;
80      float mod = ((v1 + v2 - 1) / (v1 - 1));
81      float durationFinal = (avgActivityDuration * mod * 1.5f);
82
83      StartCoroutine(InvokeAfterFrames ("finishedSitting", (int)
           durationFinal));
84    } else
85      decideNextActivity ();
86  }
87
88  void finishedSitting(){
89    controller c = GameObject.FindGameObjectWithTag ("GameController
         ").GetComponent<controller> ();
90    if (agentState == simUtils.agentStates.FeatureVisitor)
91      c.agentsFeatureVisitors -= groupSize;
92    else if (agentState == simUtils.agentStates.Sitting)
93      c.agentsSitting -= groupSize;
94    else if (agentState == simUtils.agentStates.Sports)
95      c.agentsSports -= groupSize;
96    this.transform.localScale = new Vector3(1,2,0.5f);
97    this.transform.position = locationToReturnTo;
98    agentState = simUtils.agentStates.Walking;
99    currentActivity = simUtils.agentActivities.Walk;
100   nextActivity = simUtils.agentActivities.Walk;
101   c.agentsWalking += groupSize;
102   moveToRandomLocation ();
103 }
```

Once an agent has exceeded its intended lifetime, or if it has been flagged by the controller for premature exit, it starts executing its exiting process, which is implemented through the two following functions:

```
1   void PrepareForExit(){
2     controller c = GameObject.FindGameObjectWithTag ("GameController
         ").GetComponent<controller> ();
3     if (!c.gatesExist) {
4       RemoveAgent ();
5     } else {
6       GameObject g;
7       if (c.gatesUseWeights)
8         g = simUtils.getRandomGate ();
9       else
10        g = simUtils.getRandomGateWeighed ();
11
```

```
12      agentState = simUtils.agentStates.Exiting;
13      setPathToLocation (g.transform.position);
14      g.GetComponent<gateScript> ().agentsExited ++;
15    }
16
17    c.IncreaseAgentsLeavingSoon (groupSize);
18  }
19
20  void RemoveAgent(){
21    controller c = GameObject.FindGameObjectWithTag ("GameController
          ").GetComponent<controller> ();
22    c.agents.Remove (this.gameObject);
23    c.AdjustAgentPopulation (-groupSize);
24    c.agentsWalking -= groupSize;
25    c.agentsExitingNextUpdate -= groupSize;
26    c.agentsExitingCounted -= groupSize;
27    c.calculateAgentStats (lifetime,groupSize);
28    if (debuggingTargetGO != null) {
29      Destroy (debuggingTargetGO);
30    }
31    Destroy (this.gameObject);
32  }
```

B.2 Controller Functions

The controller object is a unique entity in the simulation that takes care of higher level functions, such as agent population size, keeping track of model statistics, and reading and writing form external files. Its initialization function is as follows (note that input parameters are set from within the Unity User Interface, and are not shown here):

```
1   public void SimInit () {
2     if (saveToFile) {
3       string runParams = SetModelParamsString ();
4
5       System.Guid guid = System.Guid.NewGuid ();
6       runId = ((int)(System.DateTime.UtcNow - new System.DateTime
            (1970, 1, 1)).TotalSeconds).ToString () + "_" + guid.
            ToString ();
7
8       runId = ((int)(System.DateTime.UtcNow - new System.DateTime
            (1970, 1, 1)).TotalSeconds).ToString () + "_" + runParams;
9       Directory.CreateDirectory ("modelRuns/" + runId);
10
11      WriteModelParamsToFile ();
12    }
13
14    delay100f = new List<float> ();
15
16    if (useDataset) {
17      readData ();
18      agentMaxPopulation = populationPredicted [0];
19      agentMaxPopulationNextStep = populationPredicted [0];
20      updatesPopulation++;
21    }
22
23    FrameCountAtStart = Time.frameCount;
```

```
24    gates = simUtils.getGates ();
25    features = simUtils.getFeatures ();
26    if (gates.Length != 0)
27      gatesExist = true;
28    if (features.Length != 0)
29      featuresExist = true;
30
31    containerAgent = GameObject.FindGameObjectWithTag ("container-
          Agent").transform;
32
33    if (uiLineMaxAgents && uiLineTotalAgents &&
          uiLineMaxAgentsNoExits){
34      uiElementsExist = true;
35    }
36
37    StartCoroutine (AddAgentsOverTime (agentMaxPopulation));
38
39    if (uiElementsExist) {
40      float val = agentMaxPopulationNextStep / 20f;
41      Vector2 p = new Vector2 (updates * timeStepUiLength, val);
42      Vector2[] pts = uiLineMaxAgents.Points;
43      Vector2[] ptsNew = new Vector2[pts.Length + 2];
44      for (int i = 0; i < pts.Length; i++) {
45        ptsNew [i] = pts [i];
46      }
47      ptsNew [ptsNew.Length - 2] = p;
48
49      p = new Vector2 (updates * timeStepUiLength+timeStepUiLength,
            val);
50      ptsNew [ptsNew.Length - 1] = p;
51      uiLineMaxAgents.Points = ptsNew;
52    }
53
54    StartCoroutine(InvokeAfterFrames ("calcCurrentPop",
          updateIntervals/4));
55    StartCoroutine(InvokeAfterFrames ("ControllerUpdate",
          updateIntervals));
56  }
```

The *ControllerUpdate()* function keeps track of agent population size, and ensures that the Spatial Disaggregation Model (SDM) is not deviating from the forecast. The controller updates once every 900 frames (15 minutes in simulation time). Auxiliary functions for setting the forecast population size for the next period and also for getting the actual population during the previous step (for validation) are also included. The functions are as follows:

```
1  void ControllerUpdate(){
2    Debug.Log ("CONTROLLER UPDATING");
3    if (saveToFile) {
4      Debug.Log ("WRITING TO FILE");
5      writeModelStatsToFile ();
6    }
7    Debug.Log ("NEXT UPDATE: " + (FrameCount() + updateIntervals).
         ToString());
8
9    CalculateAgentPopulationPreviousStep ();
10   int validationDiff = agentMaxPopulationNextStep -
         agentPopDuringPrevStep;
11   calculateAgentPopulationNextStep();
12
```

```
13    int agentPopDuringNextStep = agentPopulation +
          agentsExitingCounted - agentsExitingNextUpdate;
14
15    if (agentPopDuringNextStep < agentMaxPopulationNextStep -
          validationDiff) {
16      StartCoroutine (AddAgentsOverTime (agentMaxPopulationNextStep
            - (int)(validationDiff/2f) - agentPopDuringNextStep));
17    }else if (agentPopulation + agentsExitingCounted -
          agentsExitingNextUpdate > agentMaxPopulationNextStep) {
18      RemoveAgents (agentMaxPopulationNextStep  - (int)(
            validationDiff/2f) - agentPopDuringNextStep);
19    }
20
21    agentMaxPopulation = agentMaxPopulationNextStep;
22    agentsExitingCounted = agentsExitingNextUpdate;
23
24    StartCoroutine(InvokeAfterFrames ("ControllerUpdate",
          updateIntervals));
25  }
26
27  void calculateAgentPopulationNextStep(){
28    if (!useDataset) {
29      if (usePopulationCap && agentPopulation + increaseAmt >
            populationCap) {
30        increaseMaxAgentsPerUpdate = false;
31        randomizeMaxAgents = false;
32        agentMaxPopulationNextStep = populationCap;
33      }
34      if (increaseMaxAgentsPerUpdate)
35        agentMaxPopulationNextStep += increaseAmt;
36      if (randomizeMaxAgents)
37        agentMaxPopulationNextStep += Random.Range (-randomizeAmt,
              randomizeAmt);
38    } else {
39      updatesPopulation++;
40      if (updatesPopulation >= populationPredicted.Count) {
41        Application.Quit ();
42        #if UNITY_EDITOR
43        UnityEditor.EditorApplication.isPaused = true;
44        #endif
45      } else {
46        agentMaxPopulationNextStep = populationPredicted [
              updatesPopulation];
47      }
48    }
49  }
50
51  void CalculateAgentPopulationPreviousStep(){
52    if (!useDataset) {
53      if (randomizeMaxAgents) {
54        agentPopDuringPrevStep = agentMaxPopulationNextStep + Random
              .Range (-randomizeAmt, randomizeAmt + 1);
55      } else {
56        agentPopDuringPrevStep = agentMaxPopulationNextStep;
57      }
58    } else {
59      agentPopDuringPrevStep = populationActual [updatesPopulation];
60    }
61  }
```

If the controller detects an inconsistency between the predicted agent population size (as provided by an external forecast model) and the expected agent population

size (as measured in the model) during its update, it adds or removes the number of required agents in the simulation, to conform to the prediction. This control is implemented as follows:

```
IEnumerator AddAgentsOverTime(int amt){
  float addDelayFloat = updateIntervals / (float)amt;
  int addDelay = updateIntervals / amt;
  float agentsPerUpdateFloat = (float)amt / (float)updateIntervals
     ;
  int agentsPerUpdate = (int)Mathf.Ceil (agentsPerUpdateFloat);
  Debug.Log ("AGENTS TO ADD: " + amt);
  Debug.Log ("ADD DELAY FLOAT: " + addDelayFloat);
  Debug.Log ("ADD DELAY: " + addDelay);
  Debug.Log ("Agents Per Update Float: " + agentsPerUpdateFloat);
  Debug.Log ("Agents Per Update: " + agentsPerUpdate);

  int i = 0;
  while(i < amt){
    int j = 0;
    while (j < agentsPerUpdate) {
      int groupSize;
      if (gatesExist) {
        GameObject g;
        if (gatesUseWeights)
          g = simUtils.getRandomGateWeighed ();
        else
          g = simUtils.getRandomGate ();
        AddAgent (g);
        groupSize = agents [agents.Count - 1].GetComponent<
           agentBase> ().groupSize;
      } else {
        Vector3 l;
        l = agentUtils.getRandomLongRangeTargetOnGround ();
        AddAgent (l);
        groupSize = agents [agents.Count - 1].GetComponent<
           agentBase> ().groupSize;
      }
      i += groupSize;
      j += groupSize;
    }
    yield return StartCoroutine(WaitForFrames(addDelay));
  }
}

void RemoveAgents(int amt){
  Debug.Log ("AGENTS TO REMOVE: " + amt);
  int i = 0;
  int j = 0;
  while (j < amt && i < agentPopulation){
    agentBase a = agents [i].GetComponent<agentBase> ();
    if (a.agentState == simUtils.agentStates.Walking) {
      a.timeAlive = a.lifetime;
      j += a.groupSize;
    }
    i += a.groupSize;
  }
}
```

Appendix C

Validation Material

C.1 CS1:HyP Forecast Model Validation

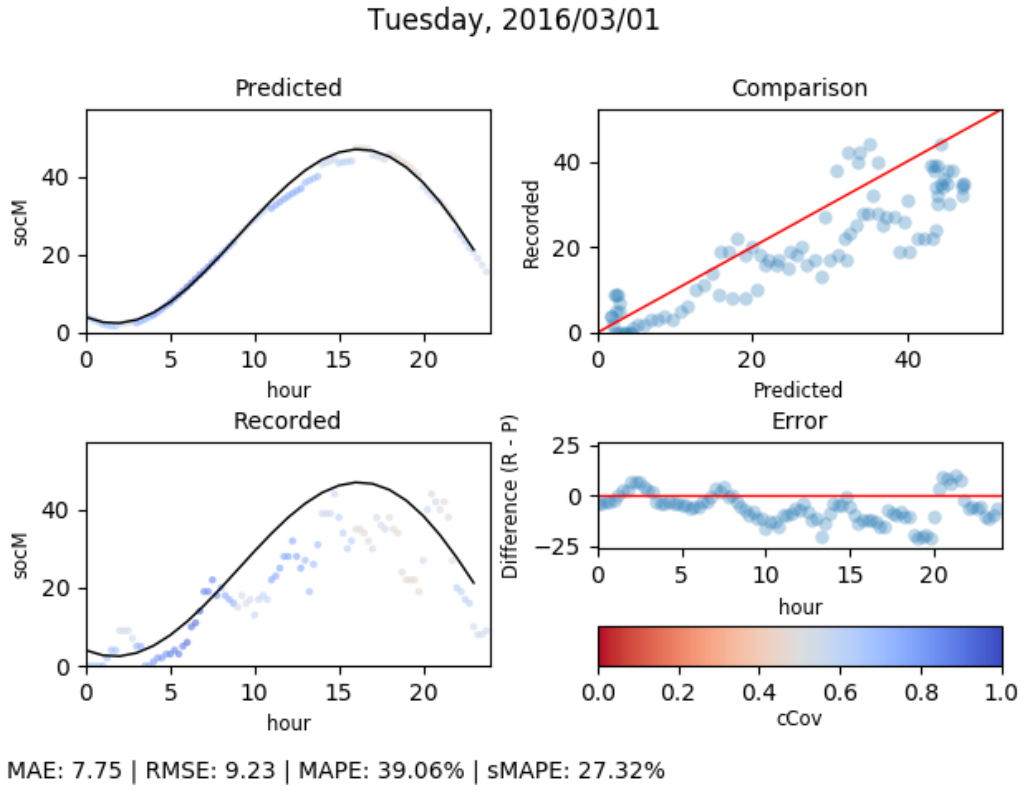


Figure C.1: CS1:HyP Forecast Model Validation for 2016-03-01

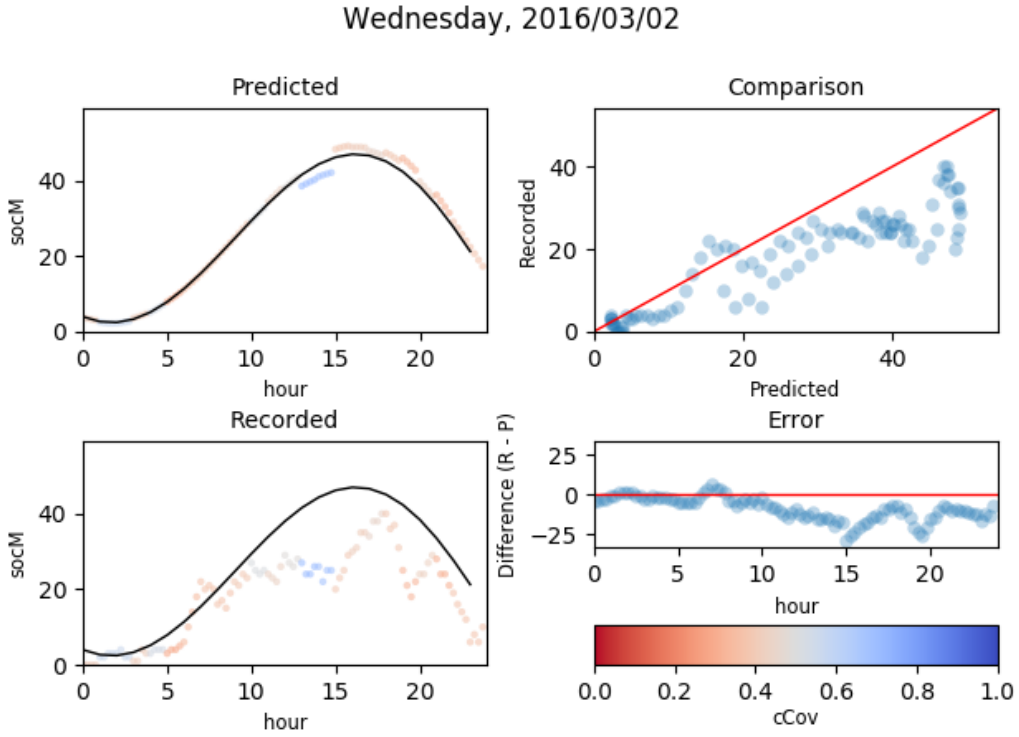


Figure C.2: CS1:HyP Forecast Model Validation for 2016-03-02

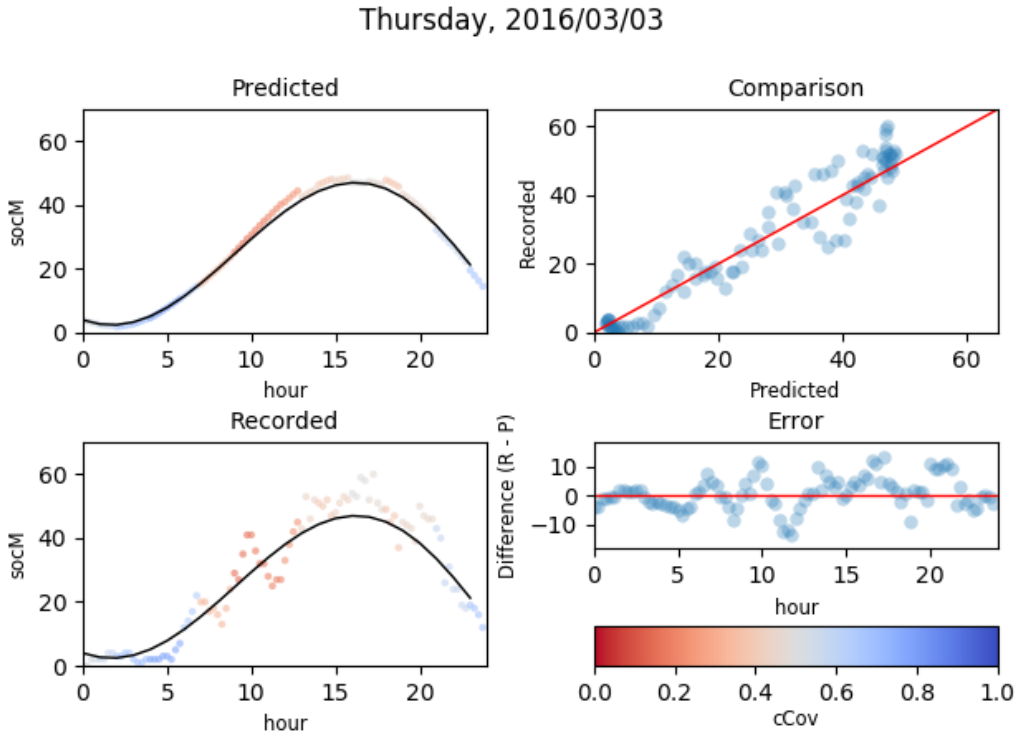


Figure C.3: CS1:HyP Forecast Model Validation for 2016-03-03

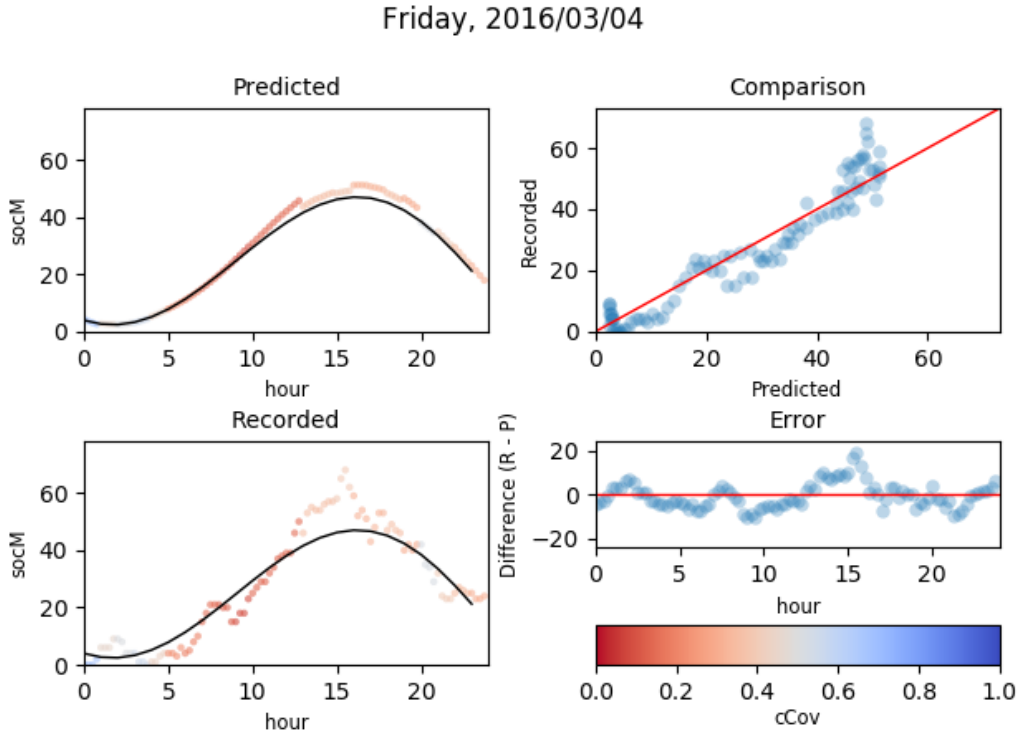


Figure C.4: CS1:HyP Forecast Model Validation for 2016-03-04

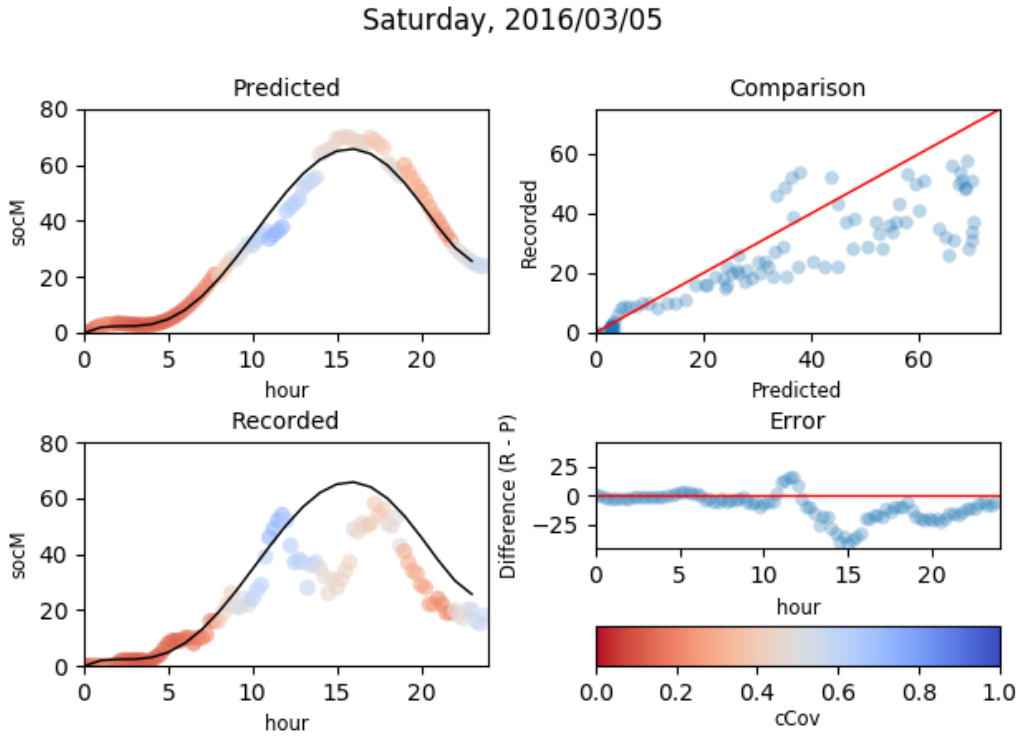
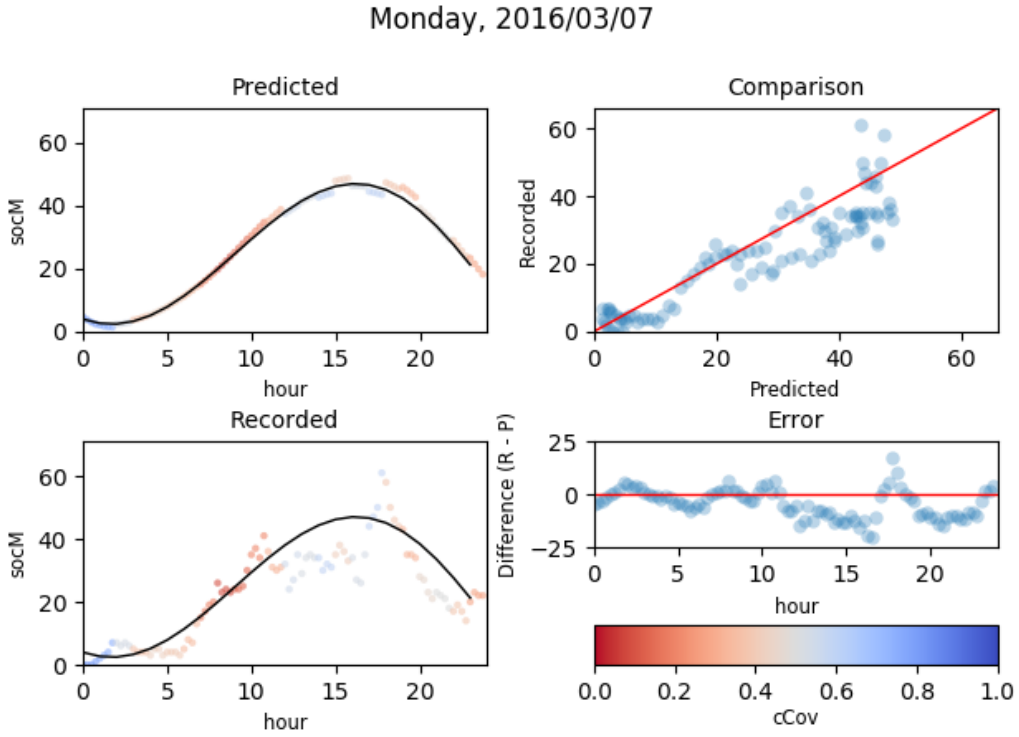
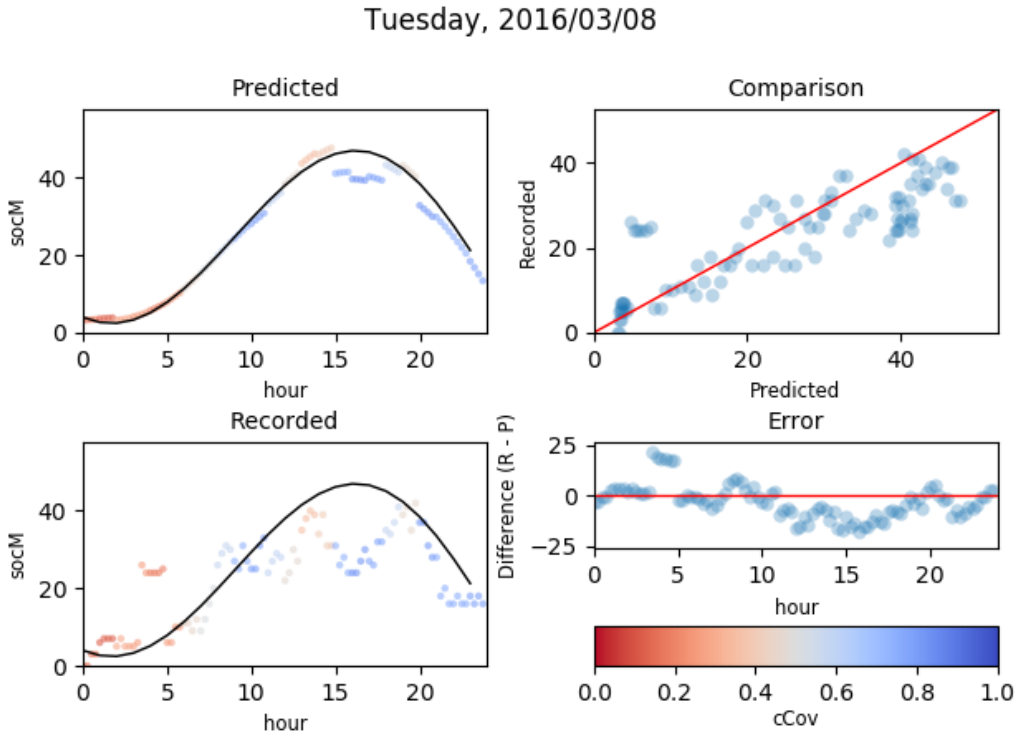


Figure C.5: CS1:HyP Forecast Model Validation for 2016-03-05



MAE: 6.22 | RMSE: 7.90 | MAPE: 27.23% | sMAPE: 18.79%

Figure C.6: CS1:HyP Forecast Model Validation for 2016-03-07



MAE: 6.71 | RMSE: 8.72 | MAPE: 29.71% | sMAPE: 17.62%

Figure C.7: CS1:HyP Forecast Model Validation for 2016-03-08

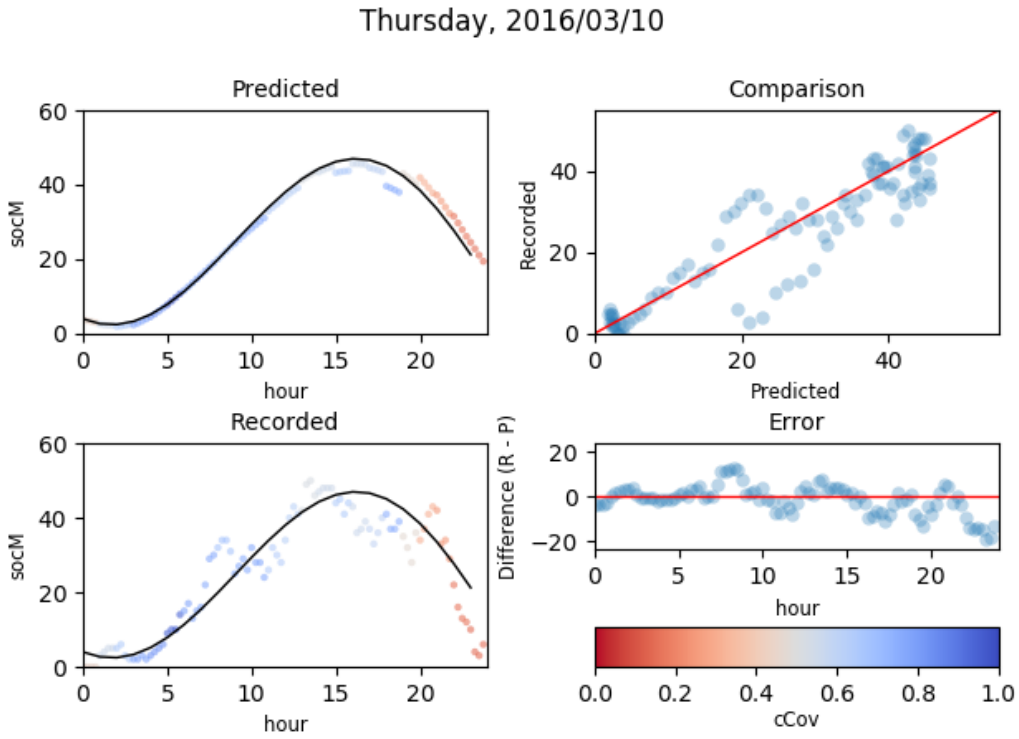


Figure C.8: CS1:HyP Forecast Model Validation for 2016-03-10

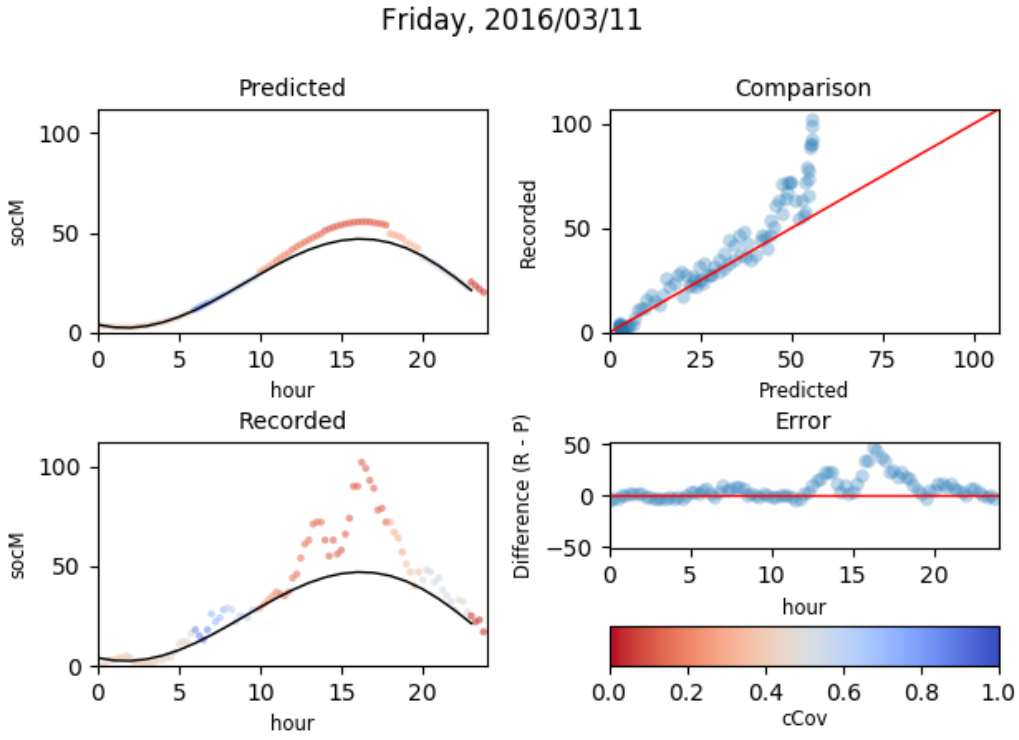


Figure C.9: CS1:HyP Forecast Model Validation for 2016-03-11

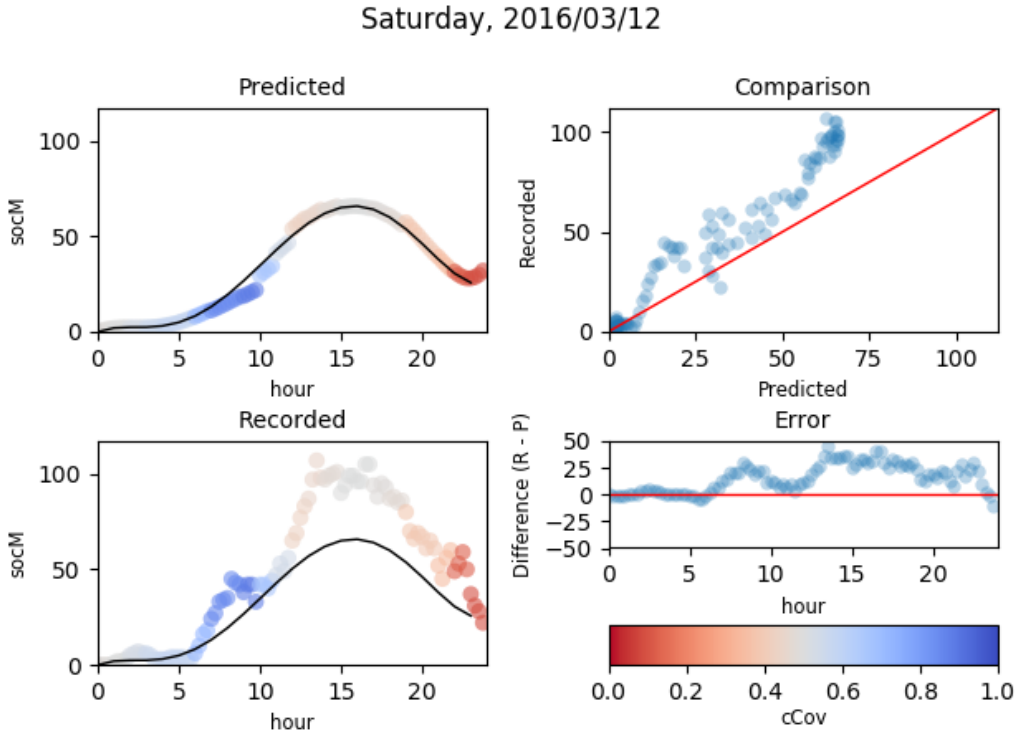


Figure C.10: CS1:HyP Forecast Model Validation for 2016-03-12

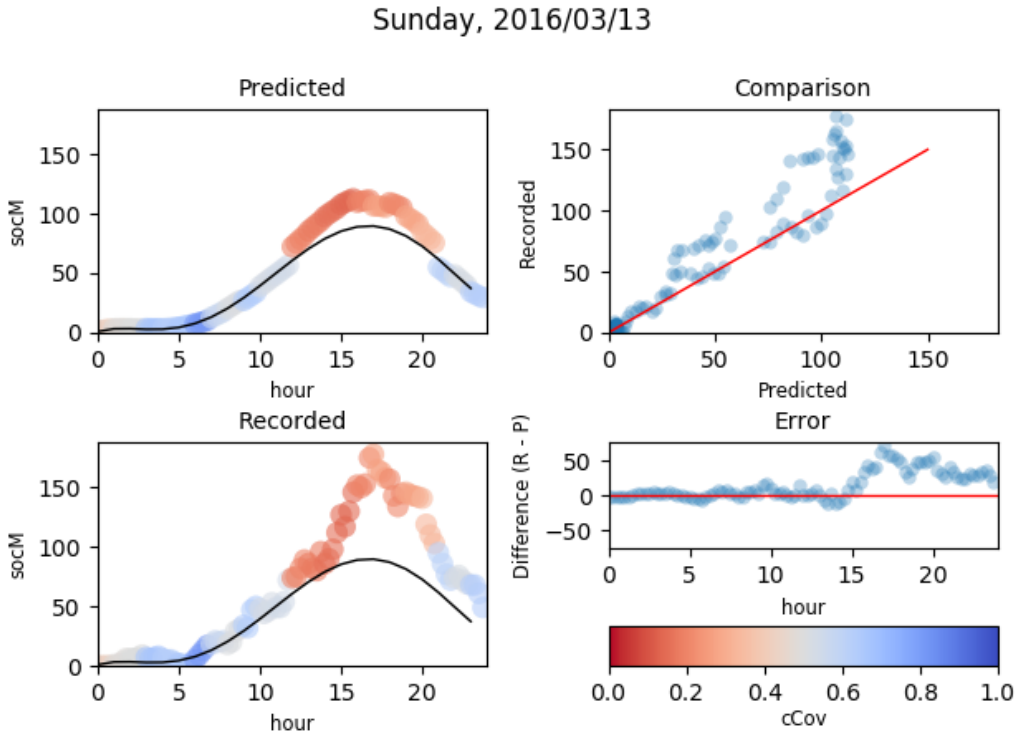
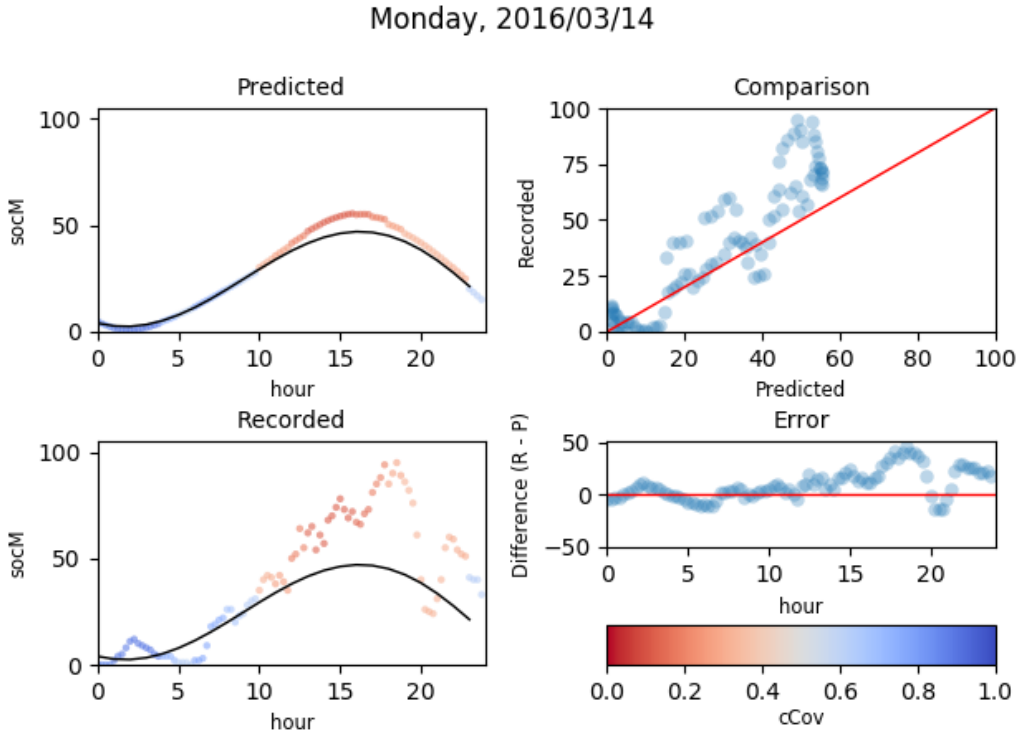
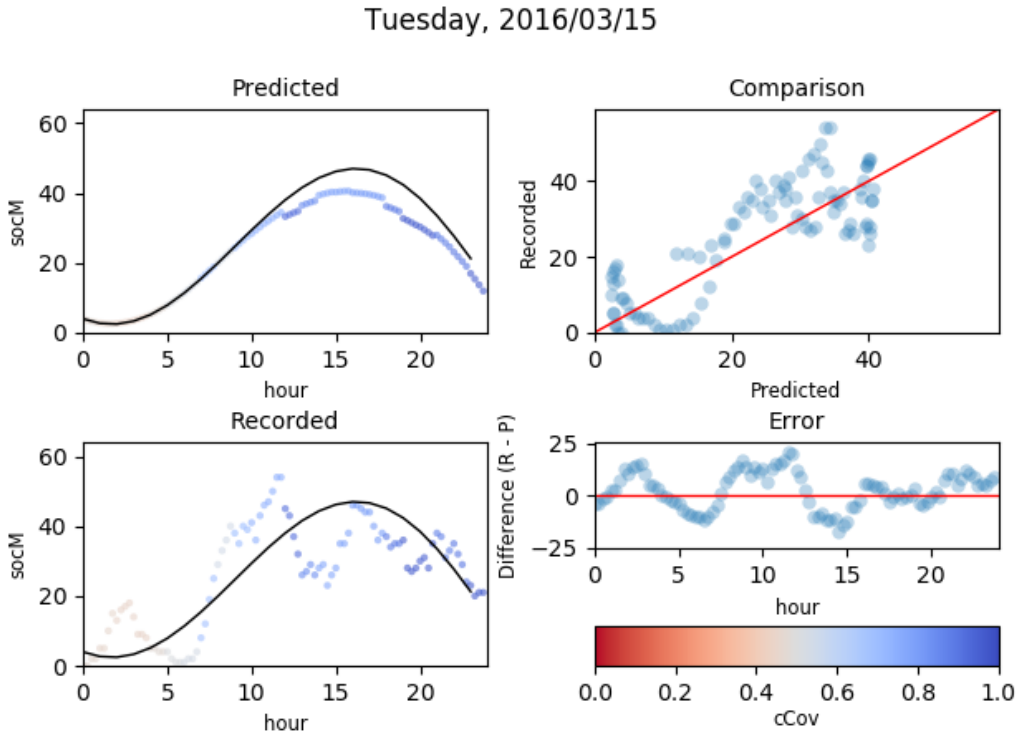


Figure C.11: CS1:HyP Forecast Model Validation for 2016-03-13



MAE: 12.84 | RMSE: 17.13 | MAPE: 33.31% | sMAPE: 29.51%

Figure C.12: CS1:HyP Forecast Model Validation for 2016-03-14



MAE: 7.56 | RMSE: 9.02 | MAPE: 29.30% | sMAPE: 24.52%

Figure C.13: CS1:HyP Forecast Model Validation for 2016-03-15

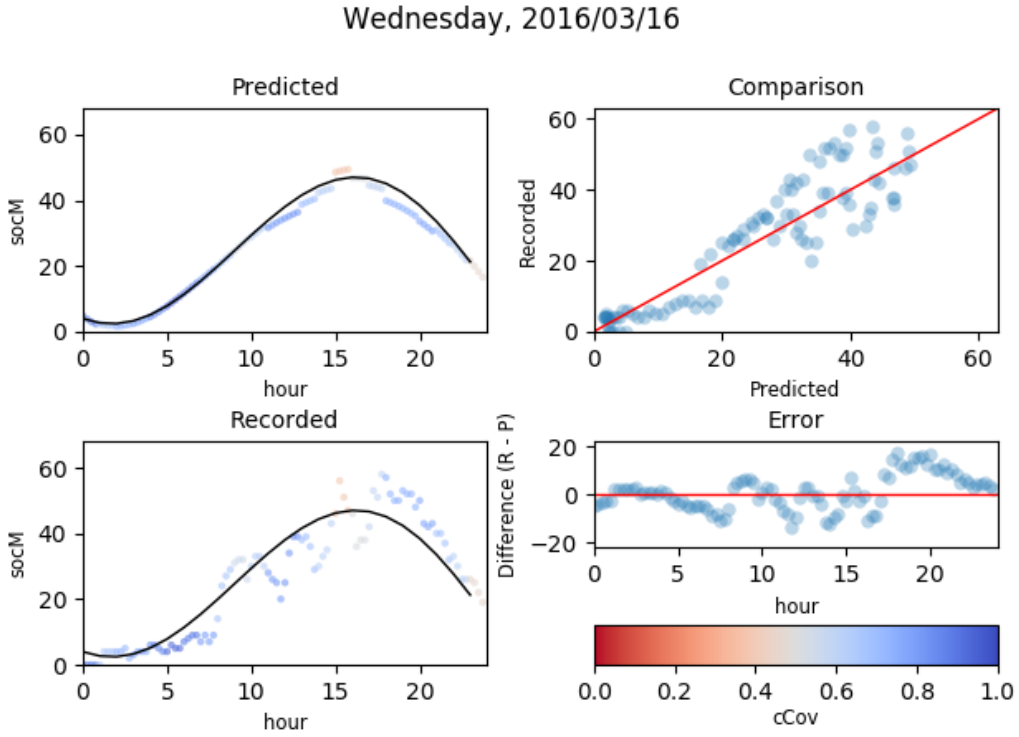


Figure C.14: CS1:HyP Forecast Model Validation for 2016-03-16

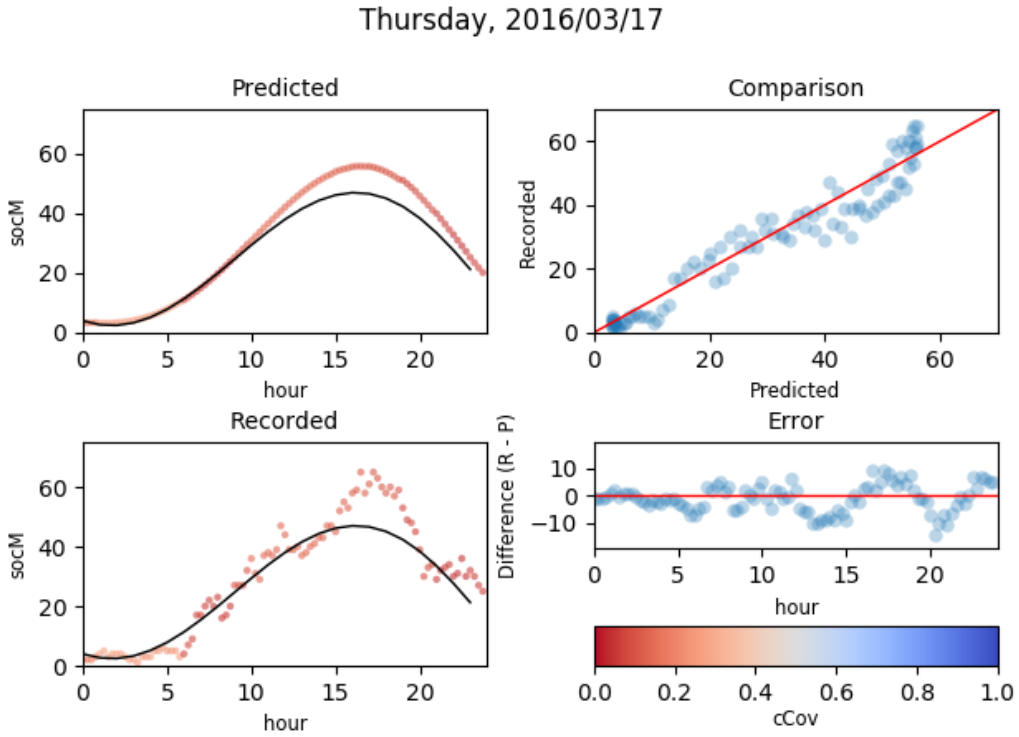
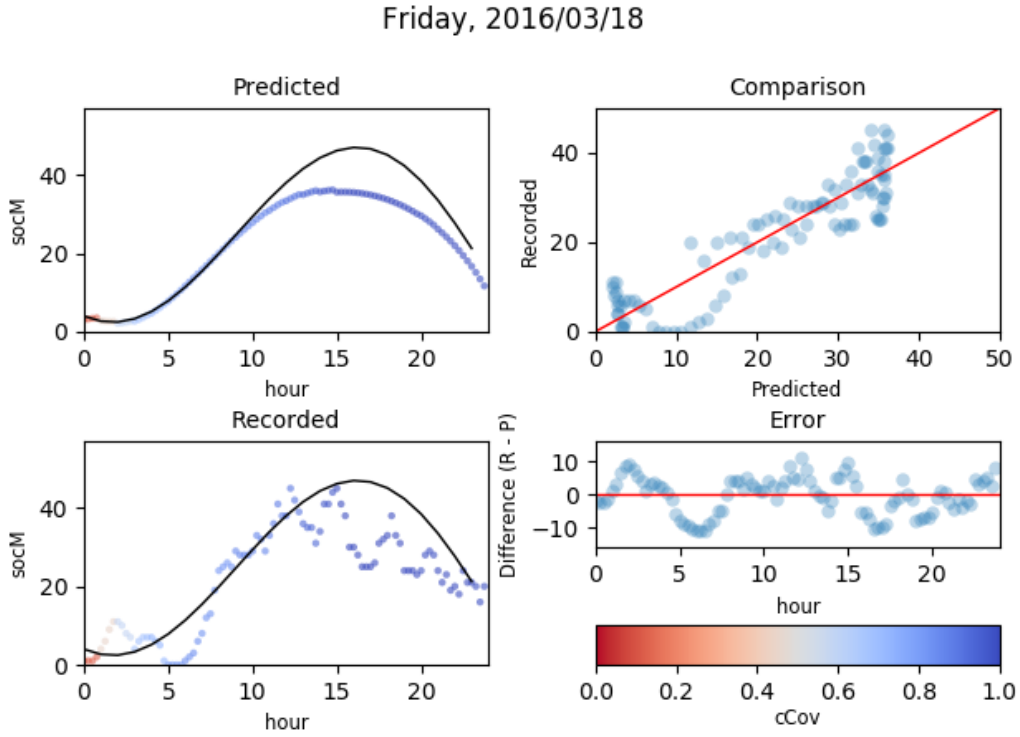
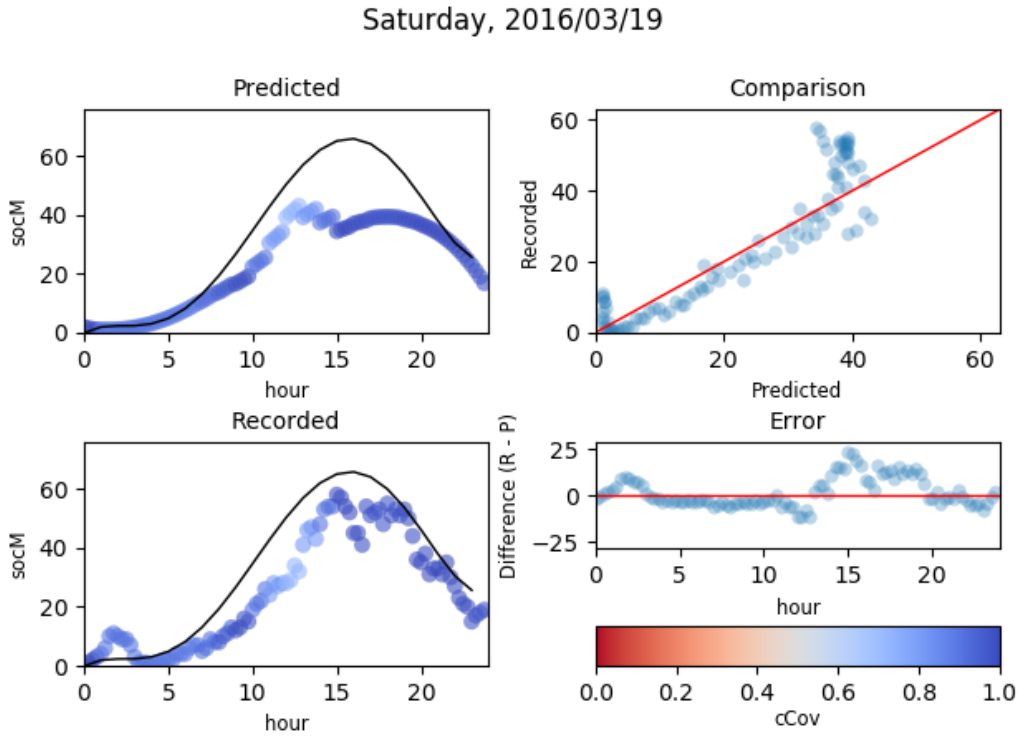


Figure C.15: CS1:HyP Forecast Model Validation for 2016-03-17



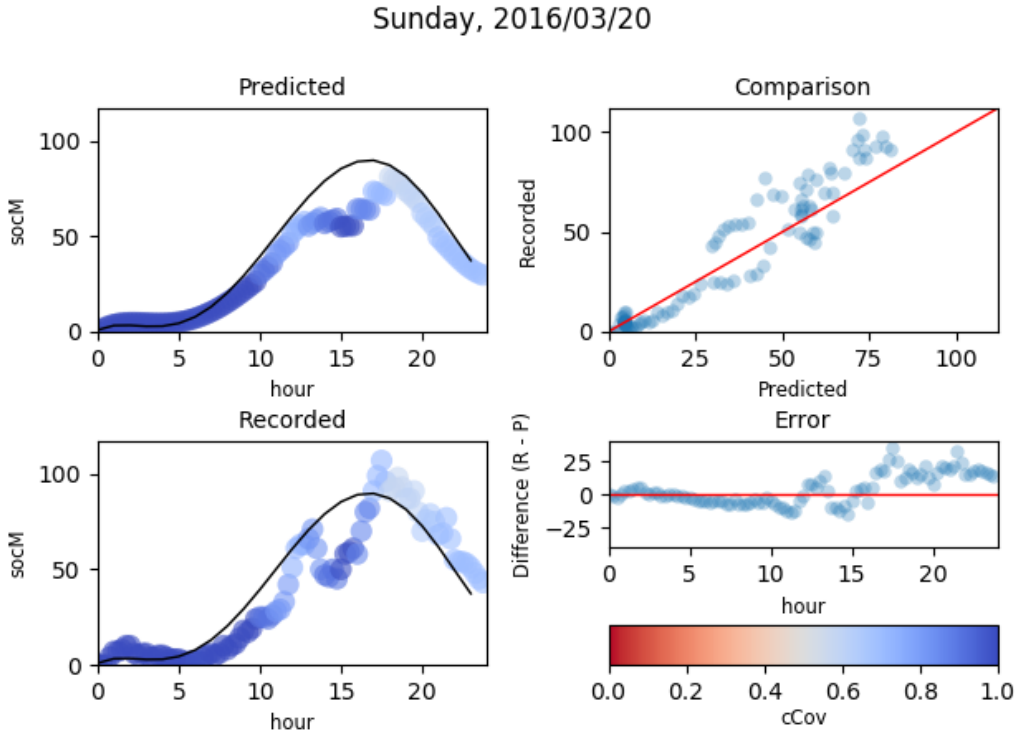
MAE: 4.64 | RMSE: 5.56 | MAPE: 21.36% | sMAPE: 21.03%

Figure C.16: CS1:HyP Forecast Model Validation for 2016-03-18



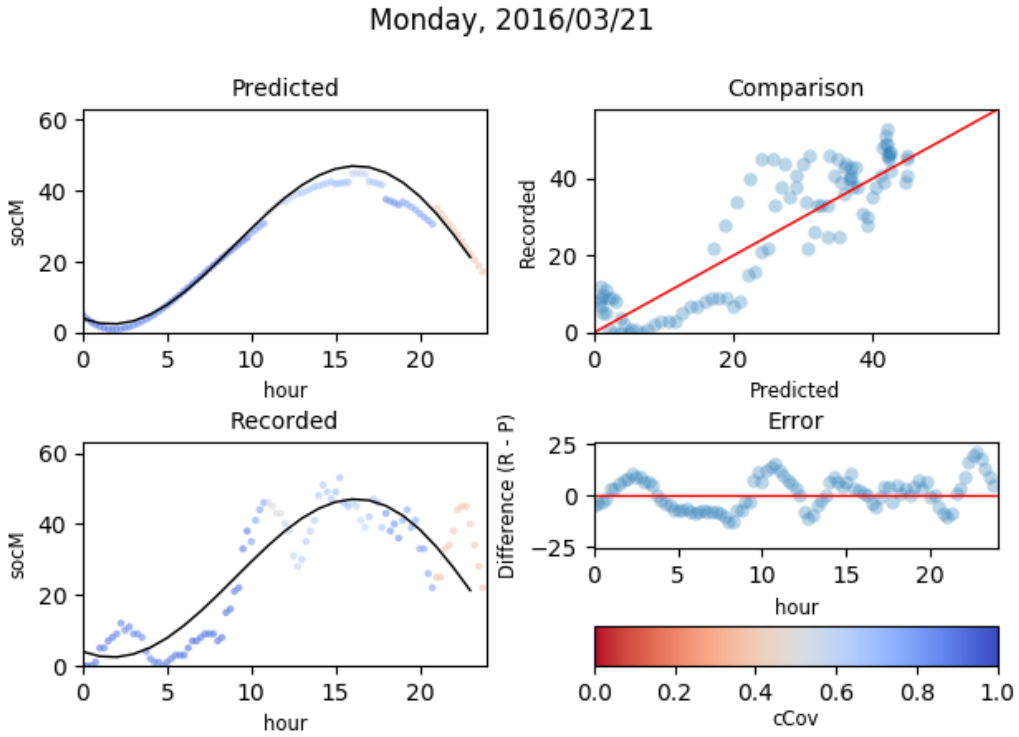
MAE: 6.20 | RMSE: 8.02 | MAPE: 25.02% | sMAPE: 22.83%

Figure C.17: CS1:HyP Forecast Model Validation for 2016-03-19



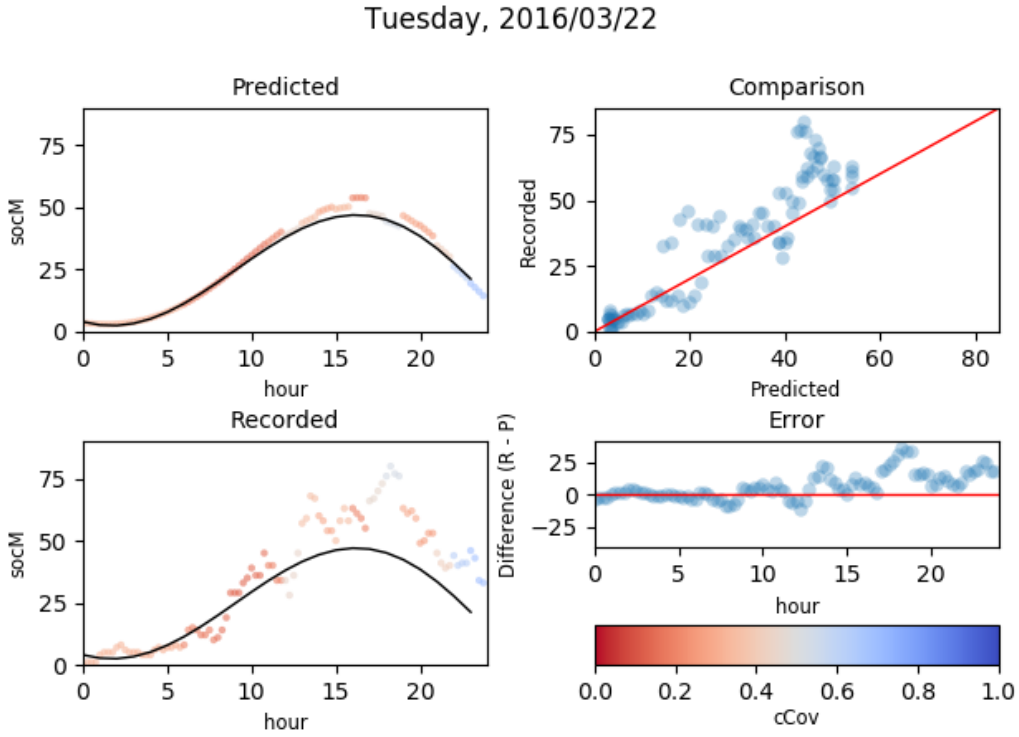
MAE: 9.24 | RMSE: 11.90 | MAPE: 23.43% | sMAPE: 18.57%

Figure C.18: CS1:HyP Forecast Model Validation for 2016-03-20



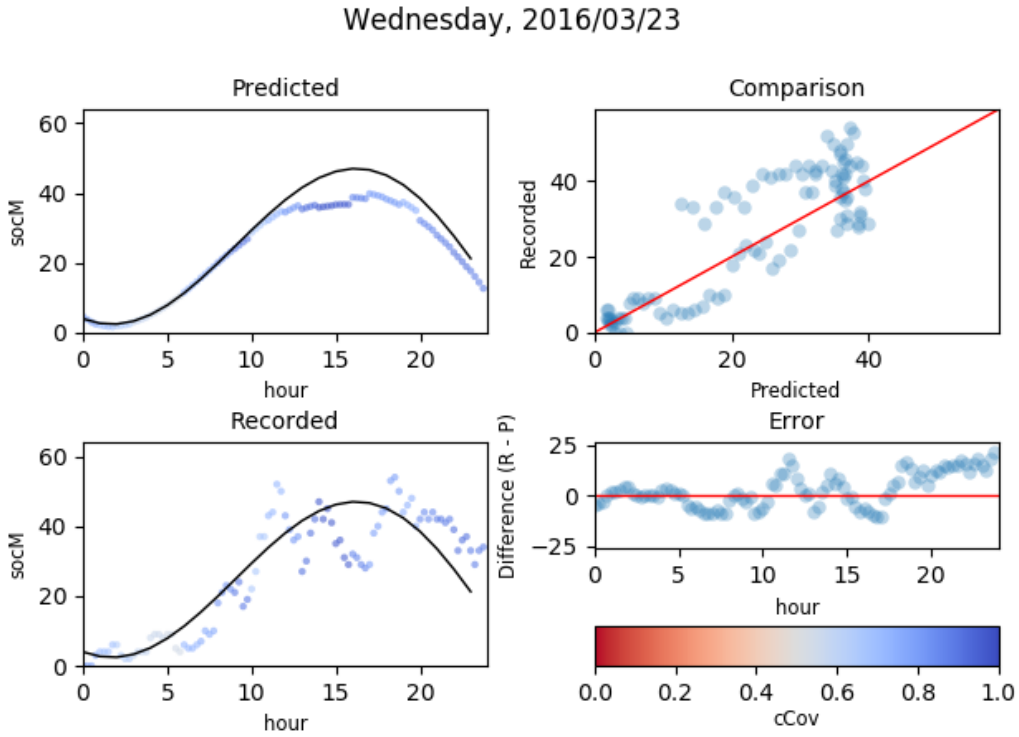
MAE: 6.69 | RMSE: 7.95 | MAPE: 25.34% | sMAPE: 27.26%

Figure C.19: CS1:HyP Forecast Model Validation for 2016-03-21



MAE: 8.87 | RMSE: 12.32 | MAPE: 25.58% | sMAPE: 16.76%

Figure C.20: CS1:HyP Forecast Model Validation for 2016-03-22



MAE: 7.03 | RMSE: 8.88 | MAPE: 26.93% | sMAPE: 19.47%

Figure C.21: CS1:HyP Forecast Model Validation for 2016-03-23

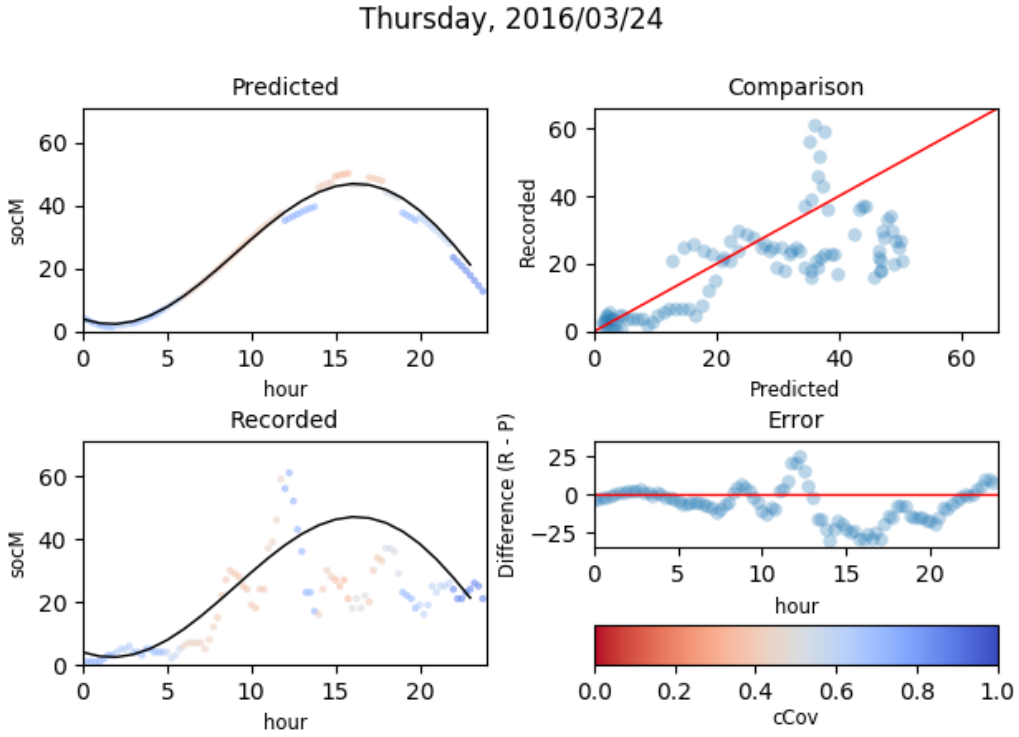


Figure C.22: CS1:HyP Forecast Model Validation for 2016-03-24

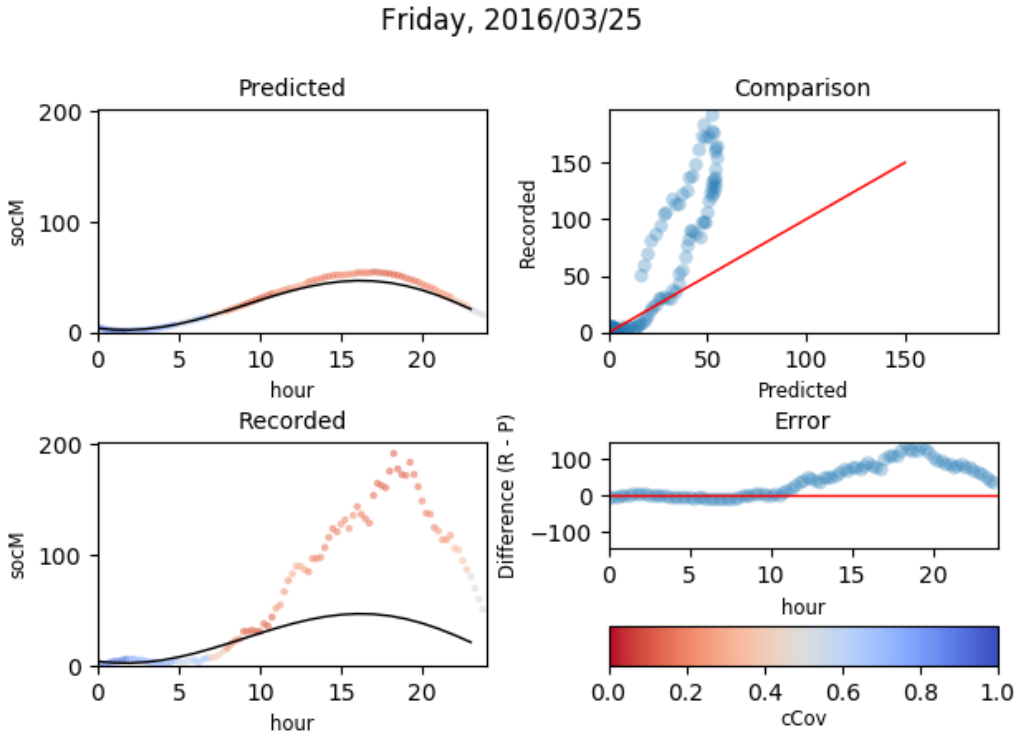
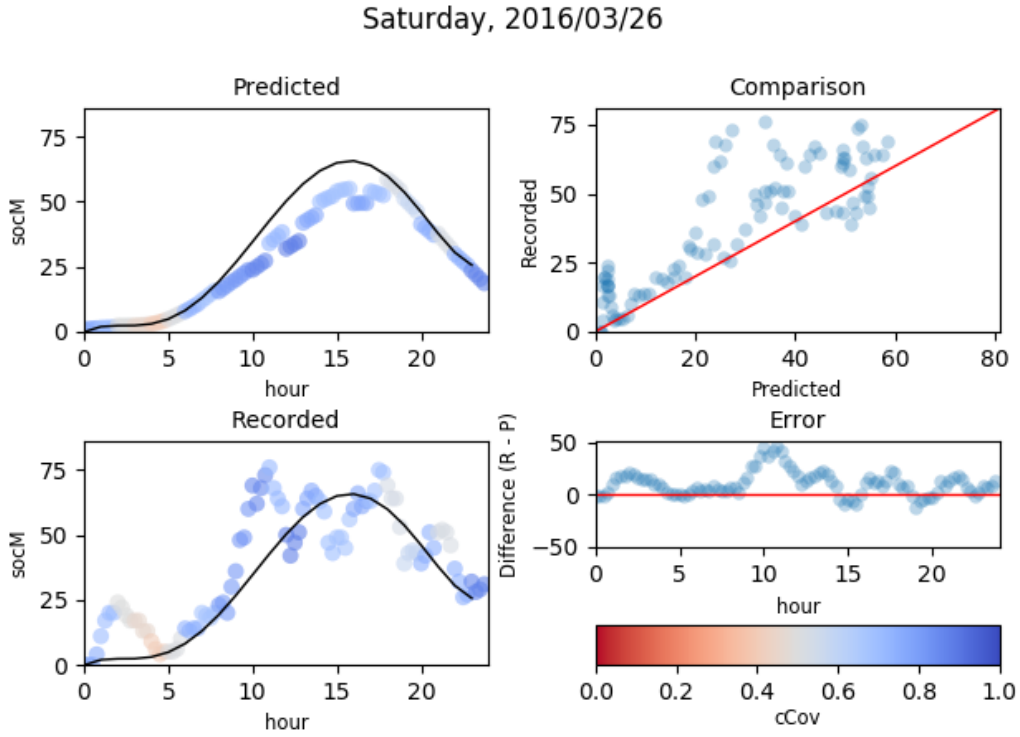
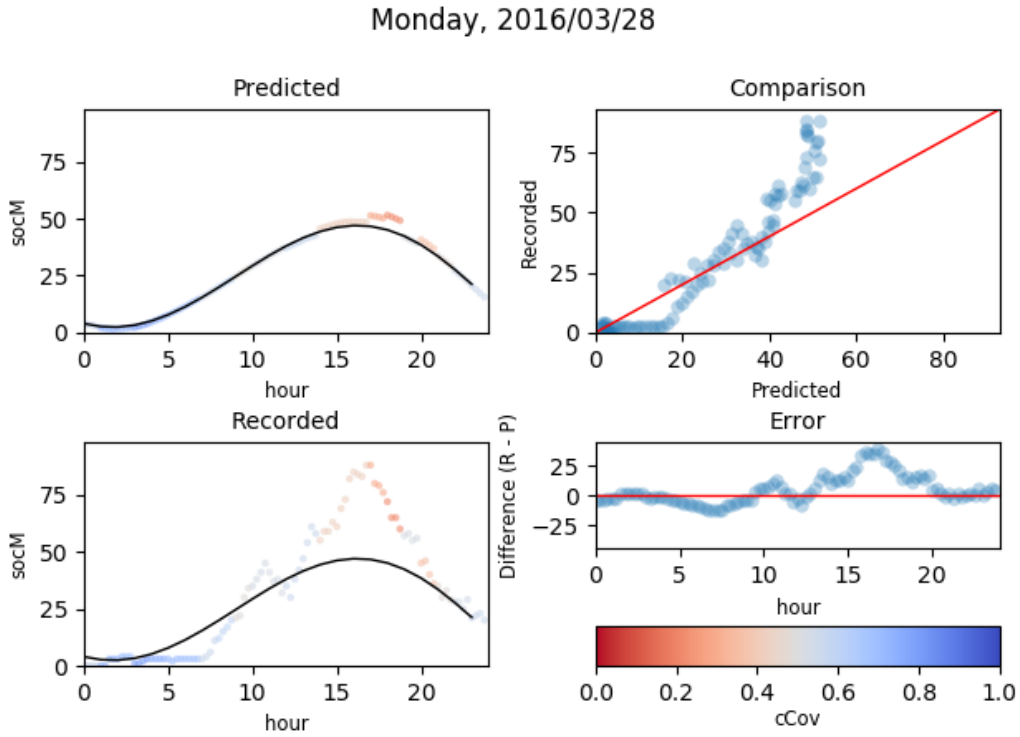


Figure C.23: CS1:HyP Forecast Model Validation for 2016-03-25



MAE: 12.19 | RMSE: 16.06 | MAPE: 32.06% | sMAPE: 27.06%

Figure C.24: CS1:HyP Forecast Model Validation for 2016-03-26



MAE: 9.23 | RMSE: 13.23 | MAPE: 28.66% | sMAPE: 25.71%

Figure C.25: CS1:HyP Forecast Model Validation for 2016-03-28

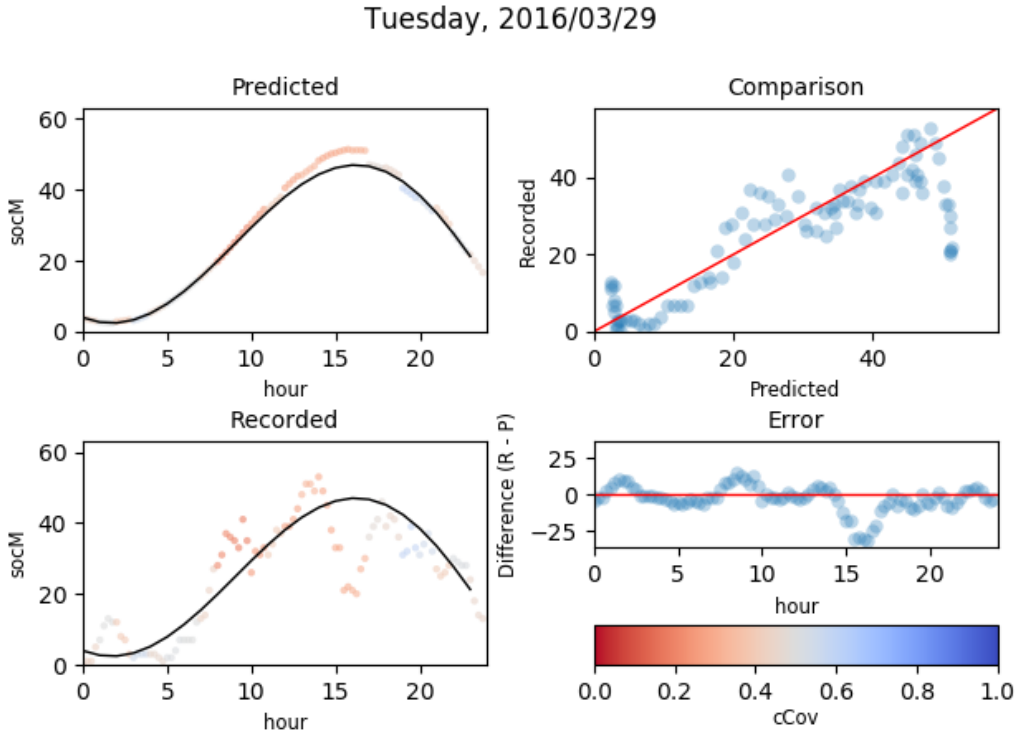


Figure C.26: CS1:HyP Forecast Model Validation for 2016-03-29

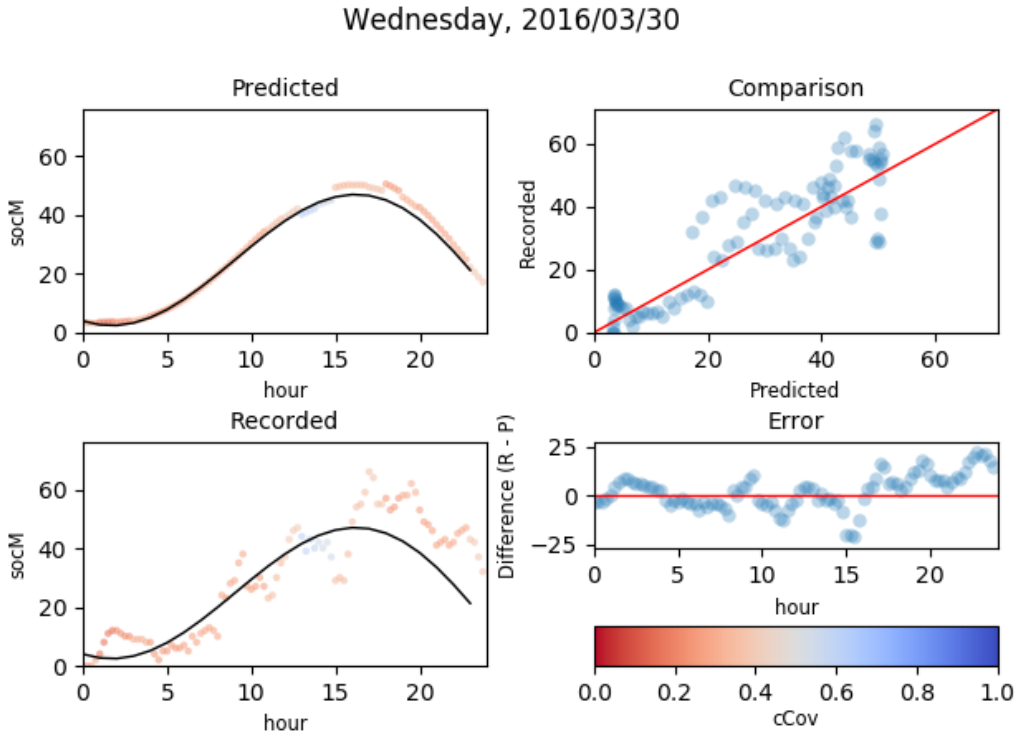


Figure C.27: CS1:HyP Forecast Model Validation for 2016-03-30

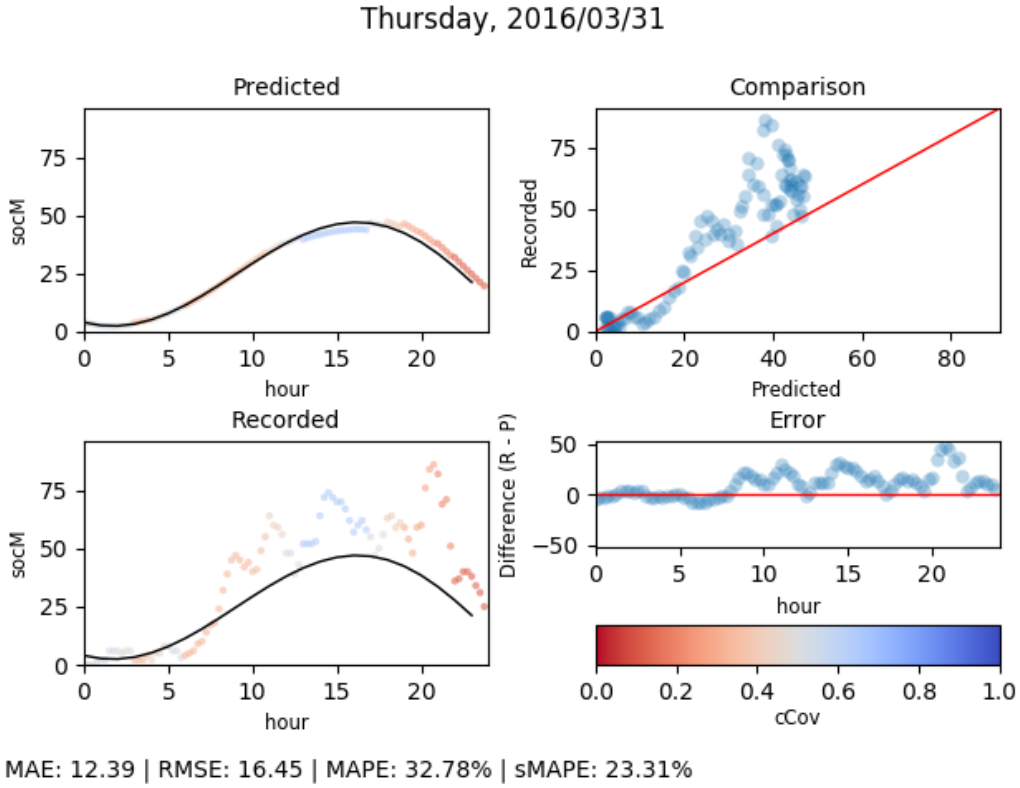


Figure C.28: CS1:HyP Forecast Model Validation for 2016-03-31

C.2 CS2:QEOP Forecast Model Validation - SocM

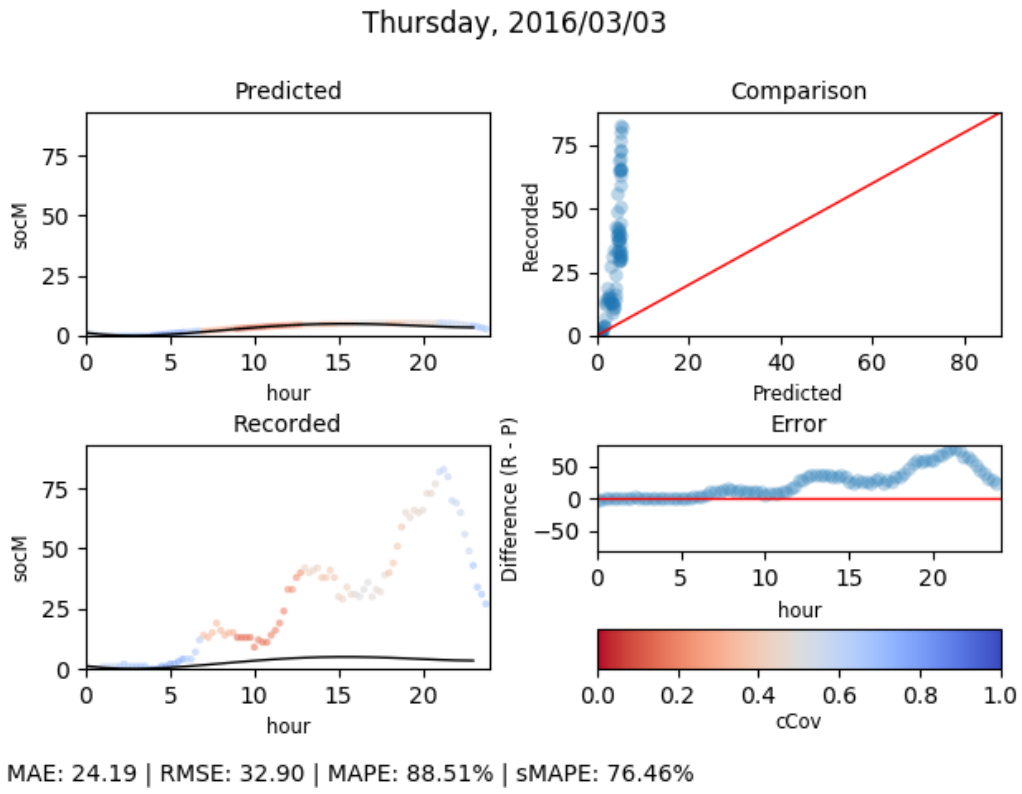
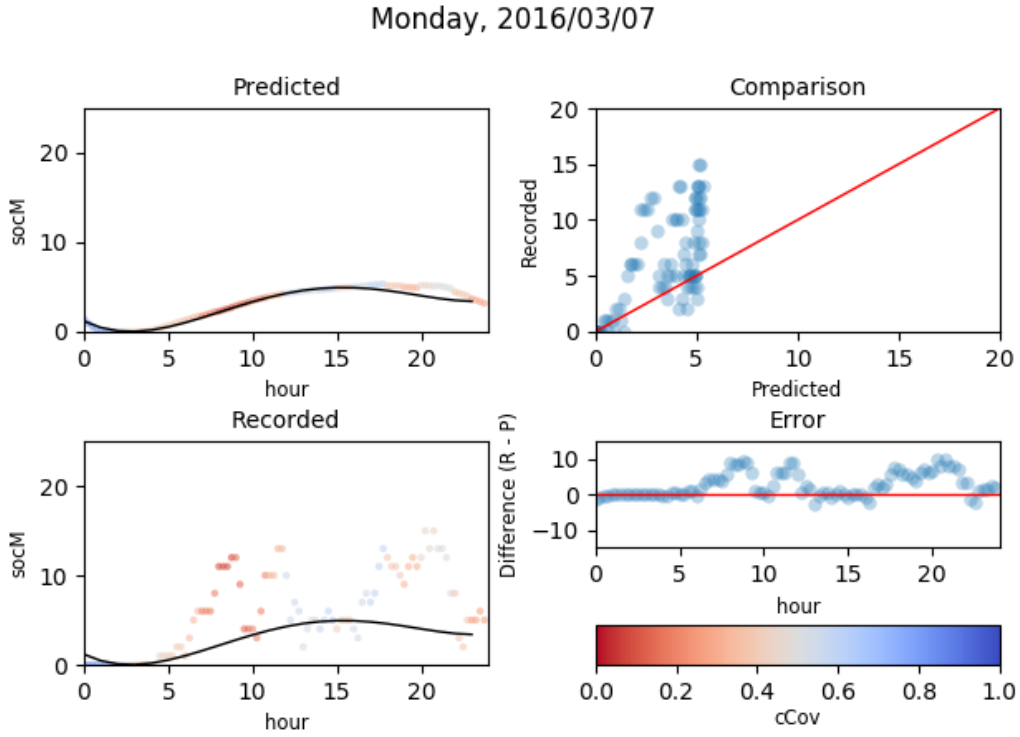
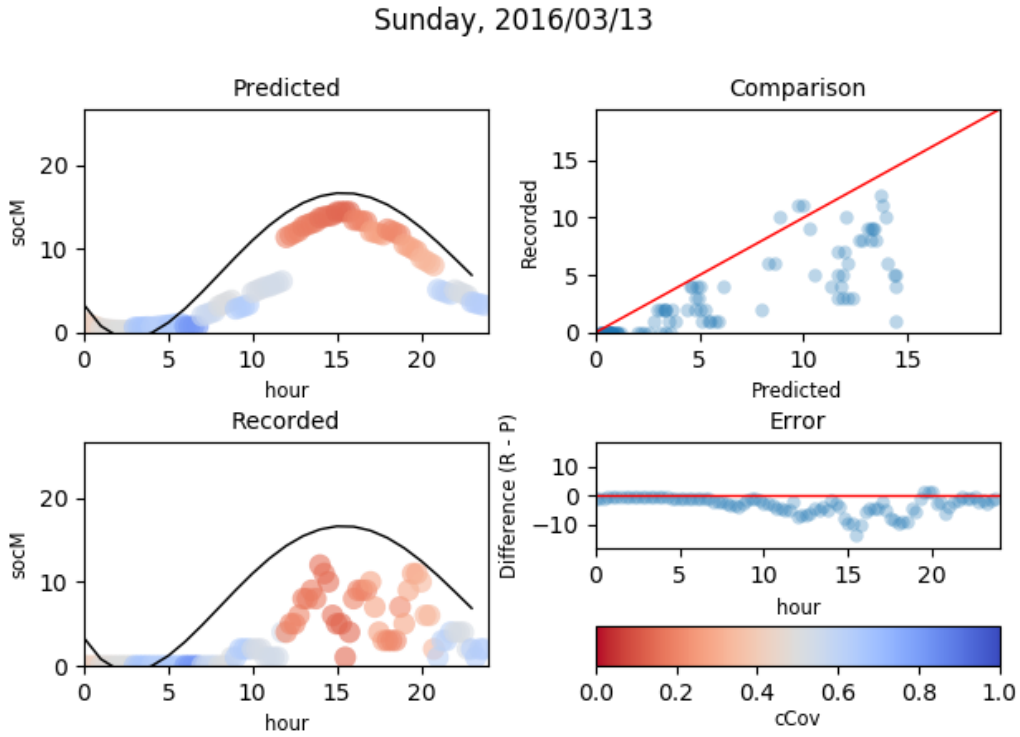


Figure C.29: CS2:QEOP SocM Forecast Model Validation for 2016-03-03



MAE: 3.04 | RMSE: 4.33 | MAPE: 51.84% | sMAPE: 32.84%

Figure C.30: CS2:QEOP SocM Forecast Model Validation for 2016-03-07



MAE: 3.08 | RMSE: 4.18 | MAPE: 100.02% | sMAPE: 58.10%

Figure C.31: CS2:QEOP SocM Forecast Model Validation for 2016-03-13

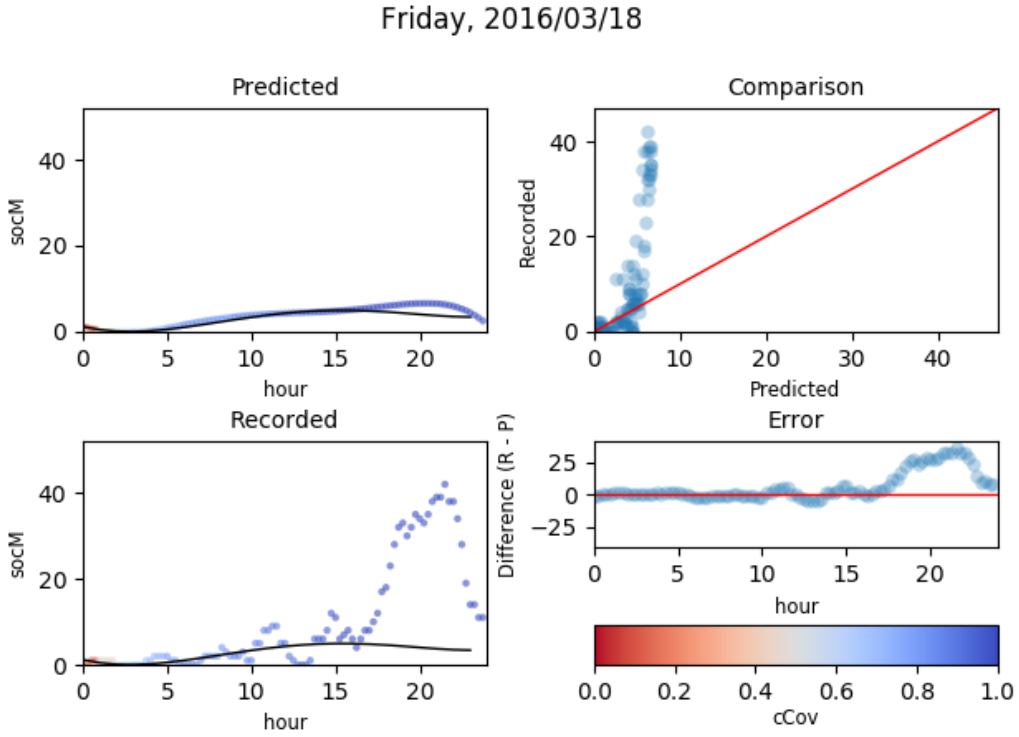


Figure C.32: CS2:QEOP SocM Forecast Model Validation for 2016-03-18

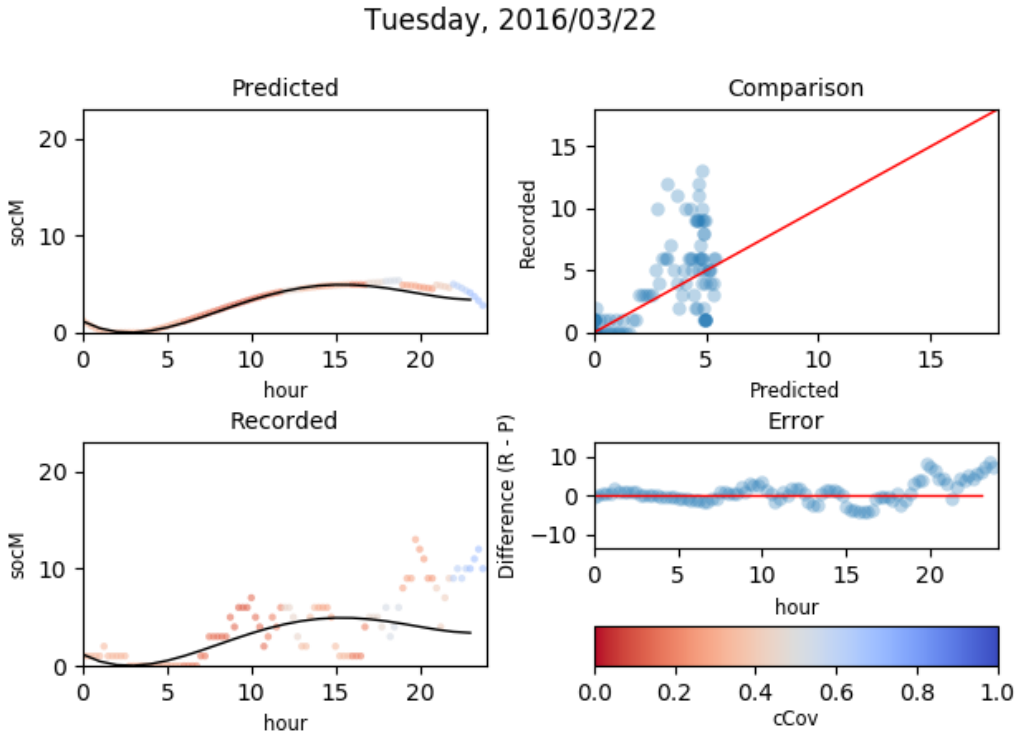


Figure C.33: CS2:QEOP SocM Forecast Model Validation for 2016-03-22

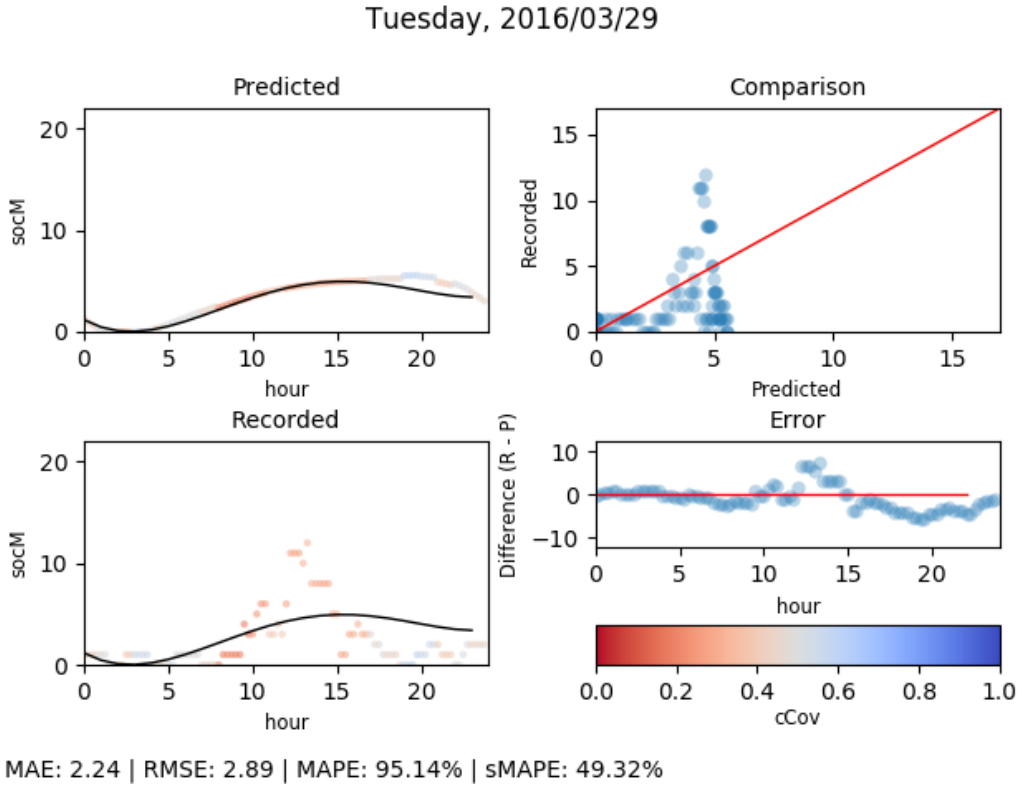


Figure C.34: CS2:QEOP SocM Forecast Model Validation for 2016-03-29

C.3 CS2:QEOP Forecast Model Validation - SocM - Naive

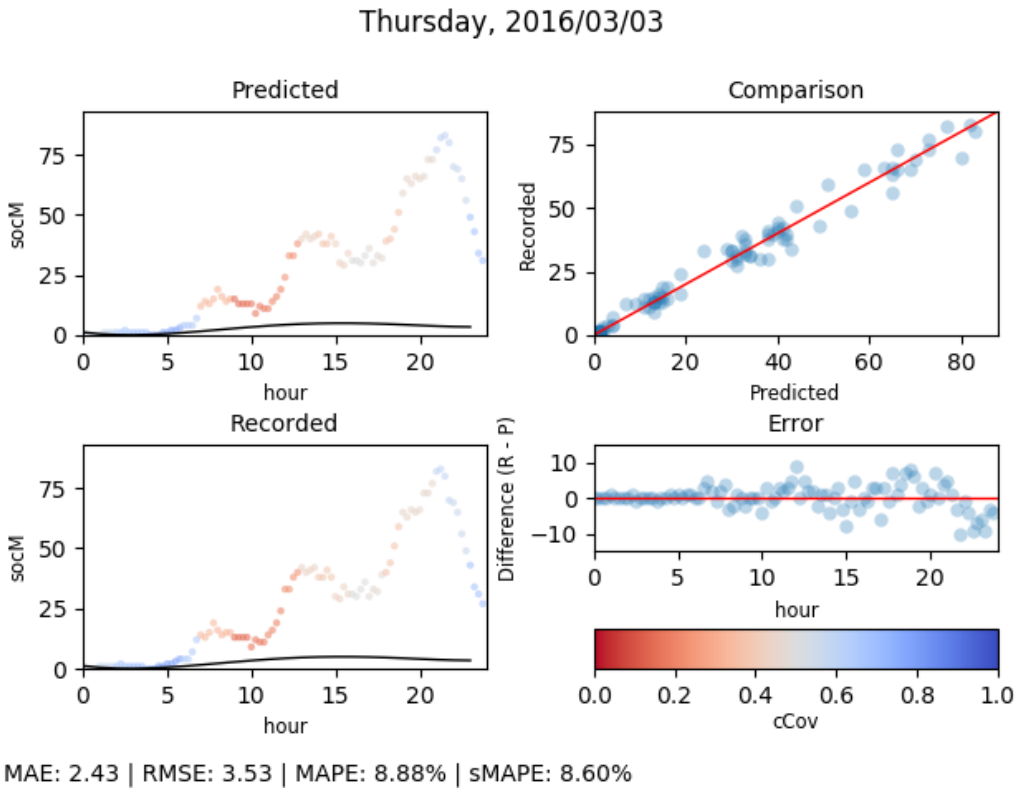
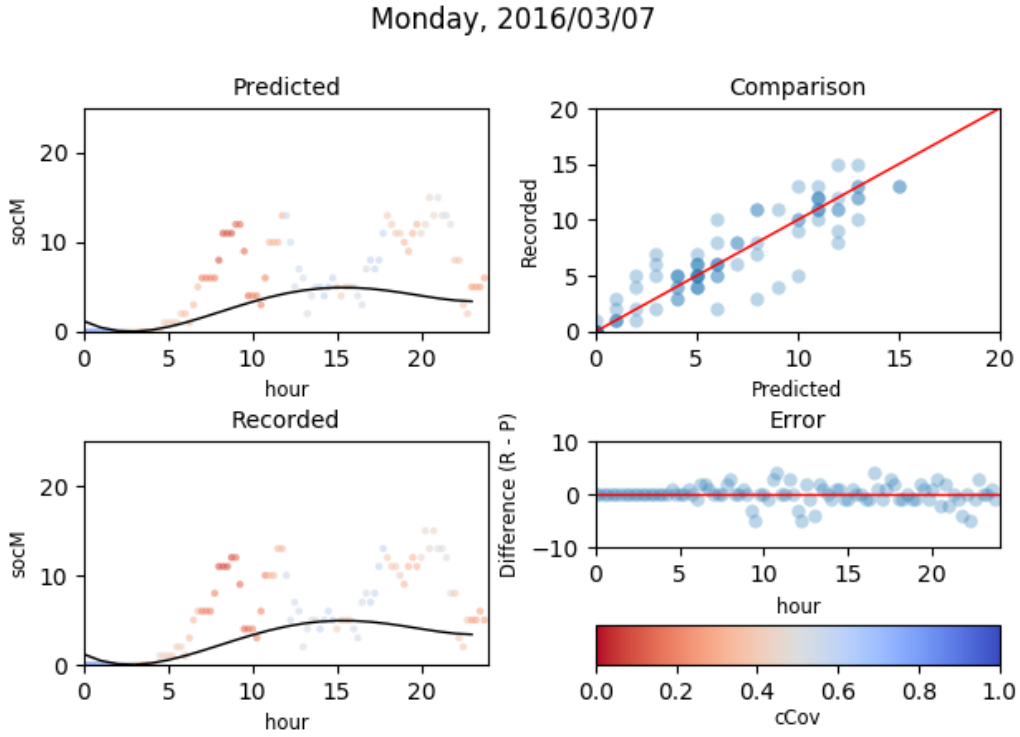
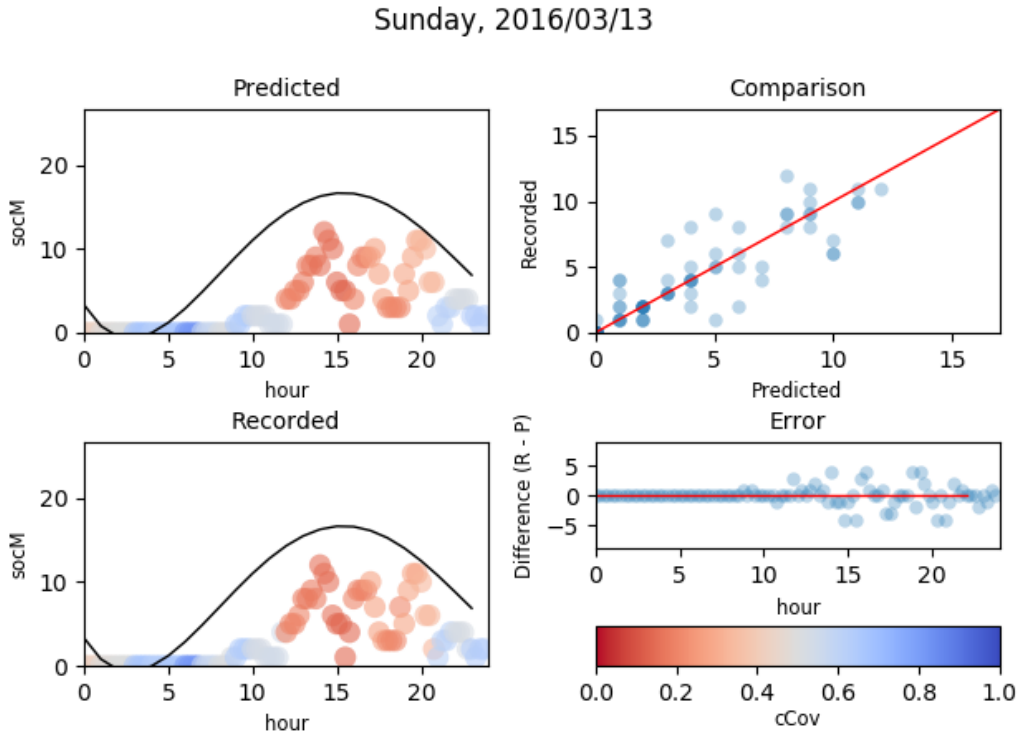


Figure C.35: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-03



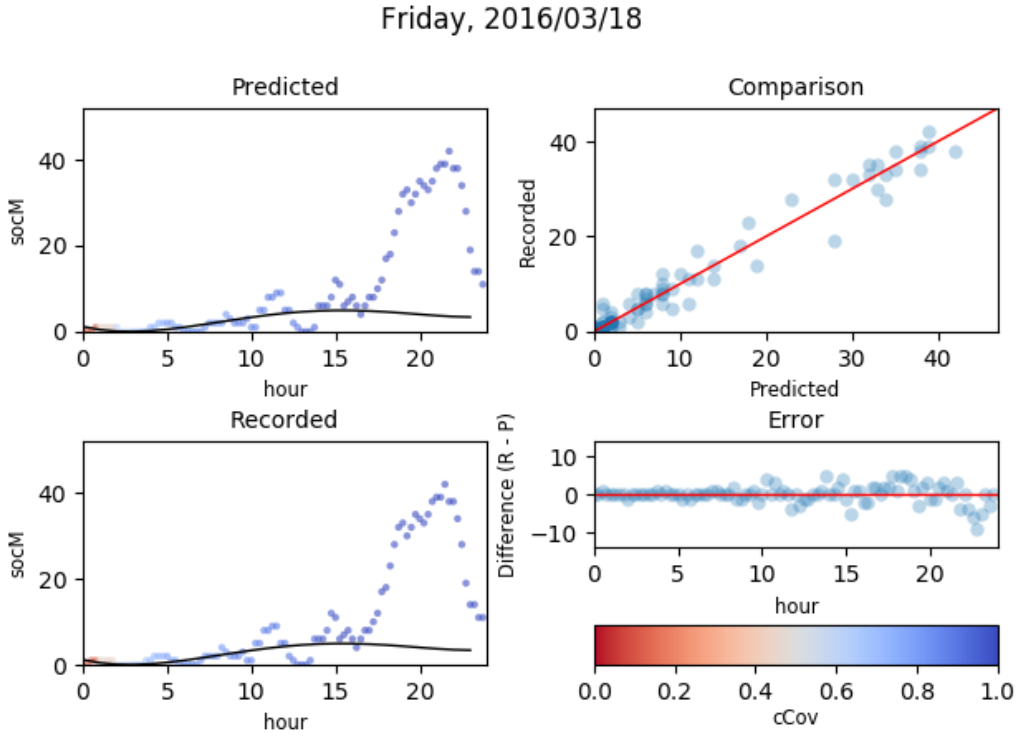
MAE: 1.09 | RMSE: 1.71 | MAPE: 18.65% | sMAPE: 10.26%

Figure C.36: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-07



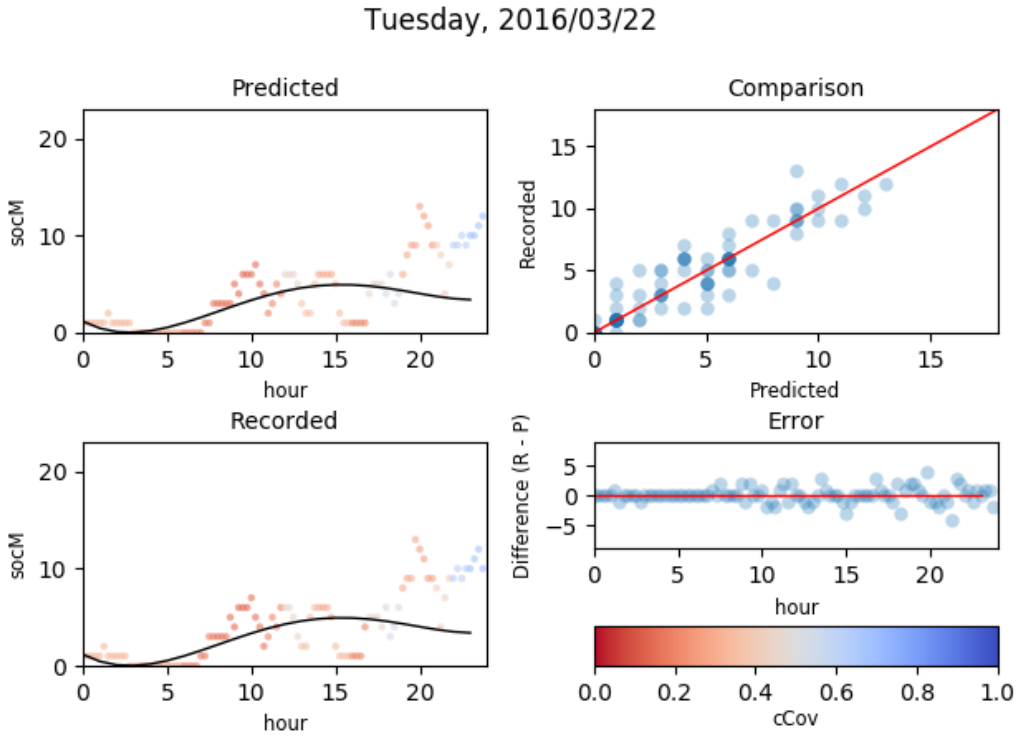
MAE: 0.75 | RMSE: 1.45 | MAPE: 24.32% | sMAPE: 9.78%

Figure C.37: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-13



MAE: 1.41 | RMSE: 2.27 | MAPE: 14.33% | sMAPE: 15.88%

Figure C.38: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-18



MAE: 0.80 | RMSE: 1.29 | MAPE: 19.79% | sMAPE: 10.92%

Figure C.39: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-22

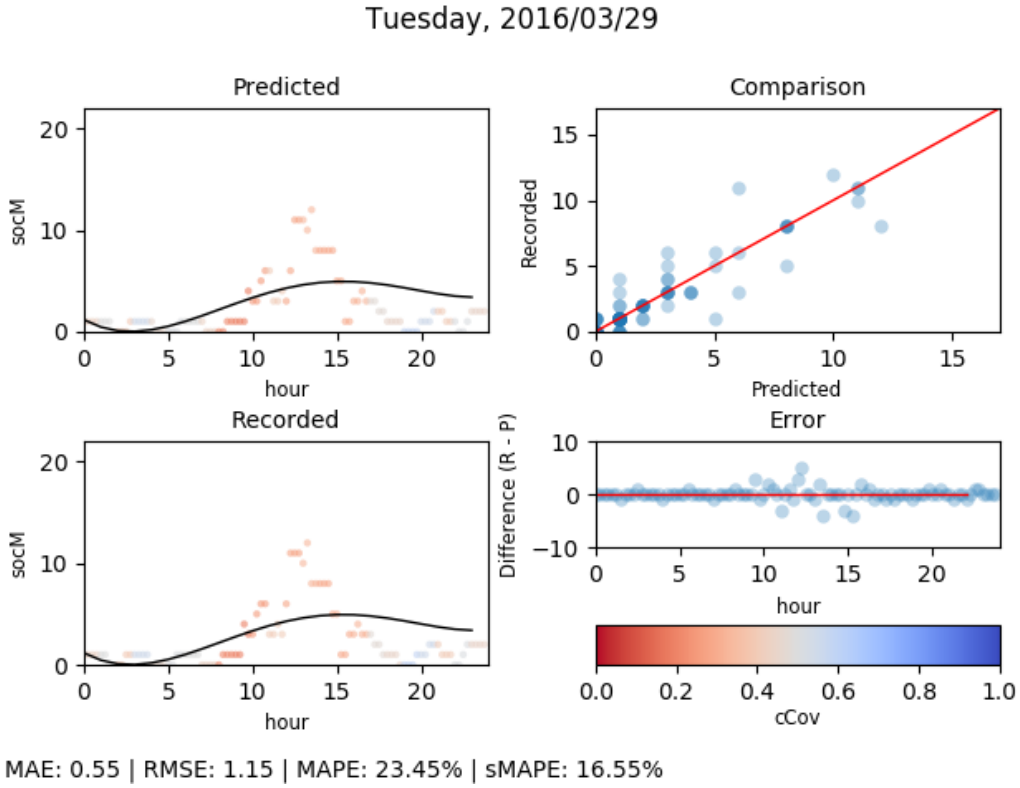


Figure C.40: CS2:QEOP SocM Naive Forecast Model Validation for 2016-03-29

C.4 CS2:QEOP Forecast Model Validation - WiFi

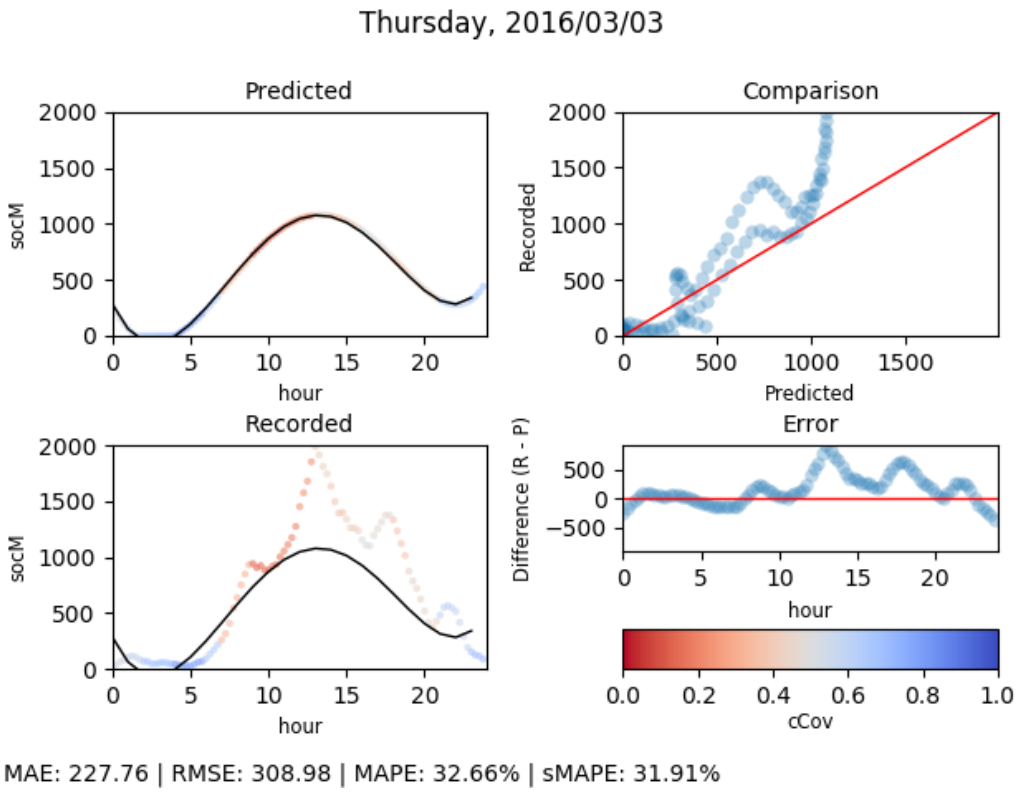
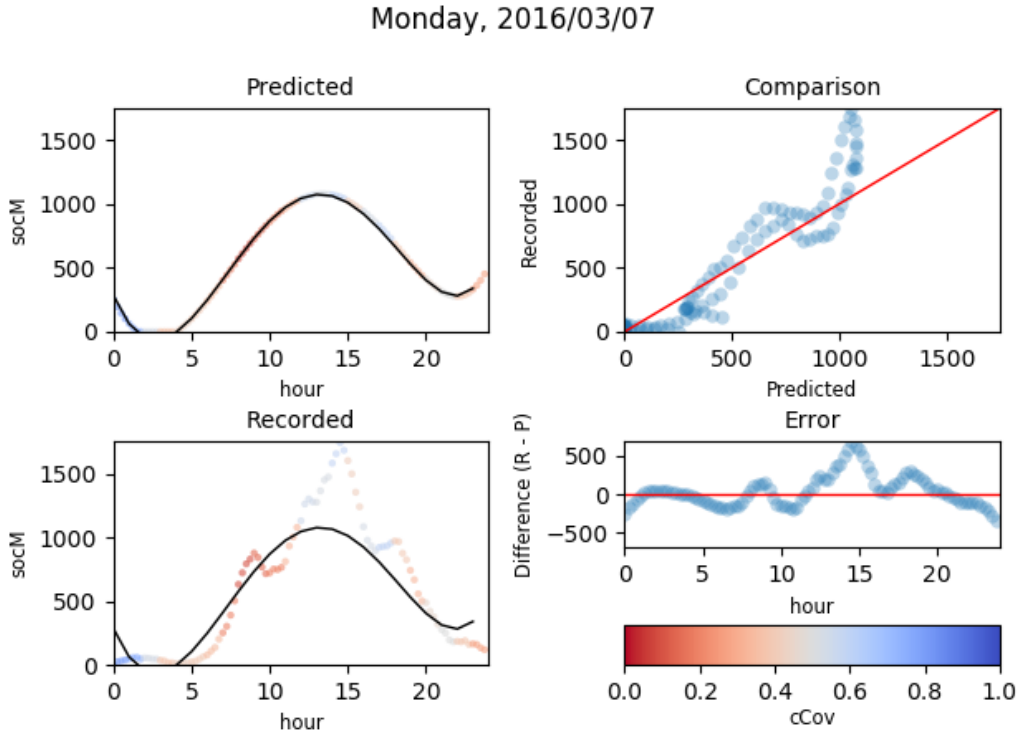
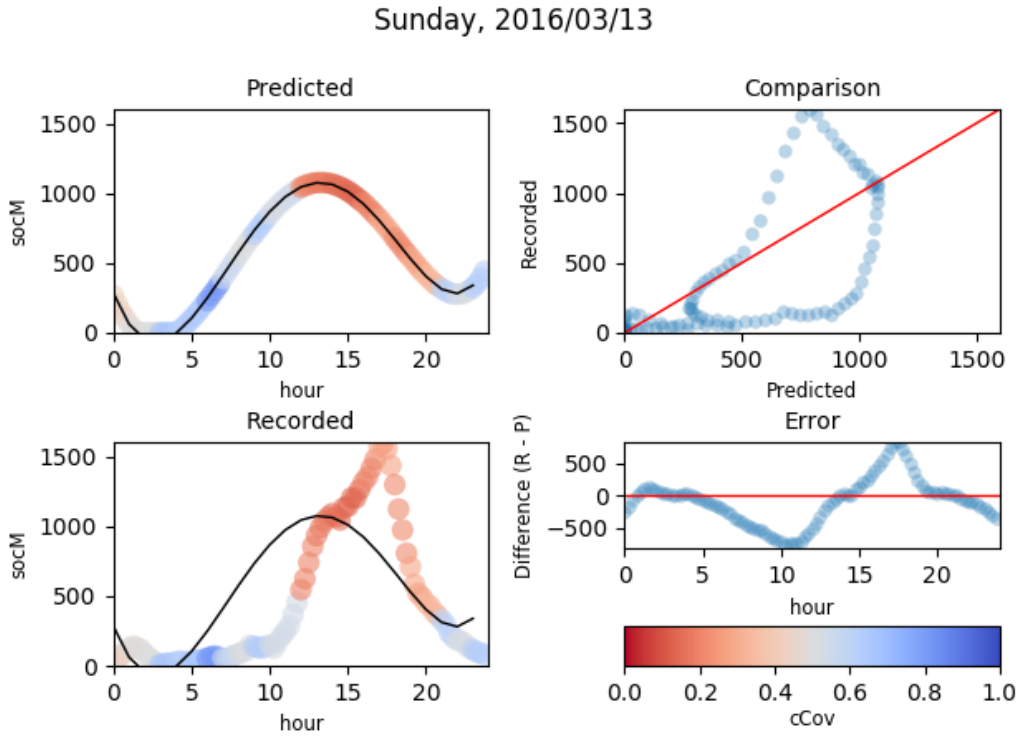


Figure C.41: CS2:QEOP WiFi Forecast Model Validation for 2016-03-03



MAE: 154.41 | RMSE: 210.34 | MAPE: 26.40% | sMAPE: 30.09%

Figure C.42: CS2:QEOP WiFi Forecast Model Validation for 2016-03-07



MAE: 271.40 | RMSE: 368.43 | MAPE: 61.02% | sMAPE: 43.76%

Figure C.43: CS2:QEOP WiFi Forecast Model Validation for 2016-03-13

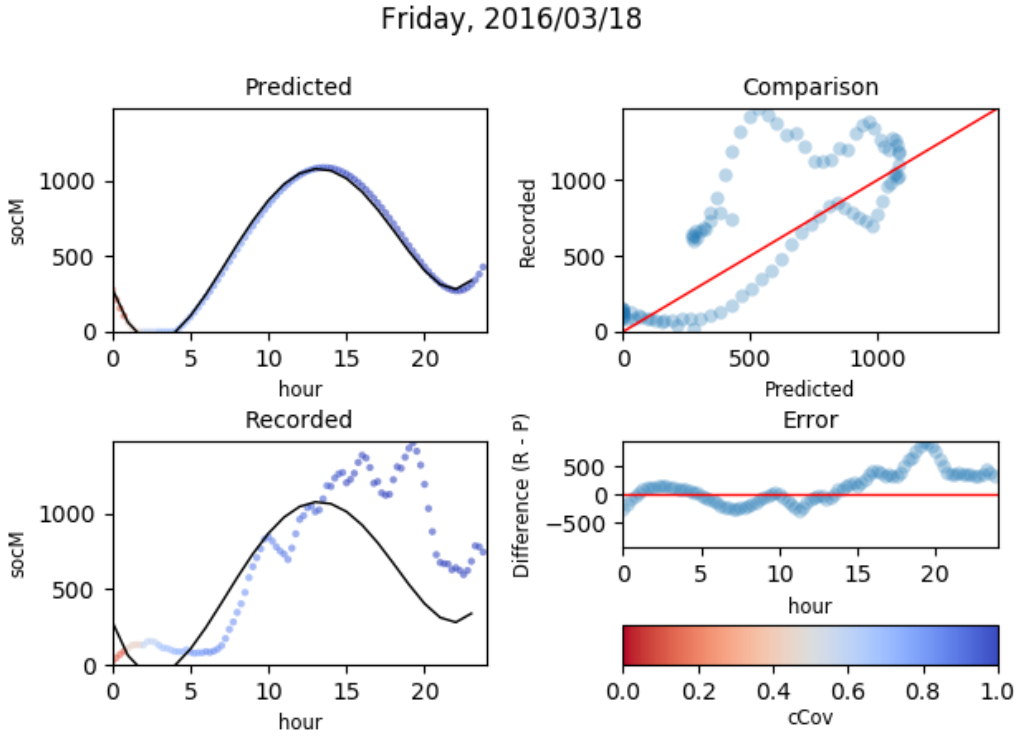


Figure C.44: CS2:QEOP WiFi Forecast Model Validation for 2016-03-18

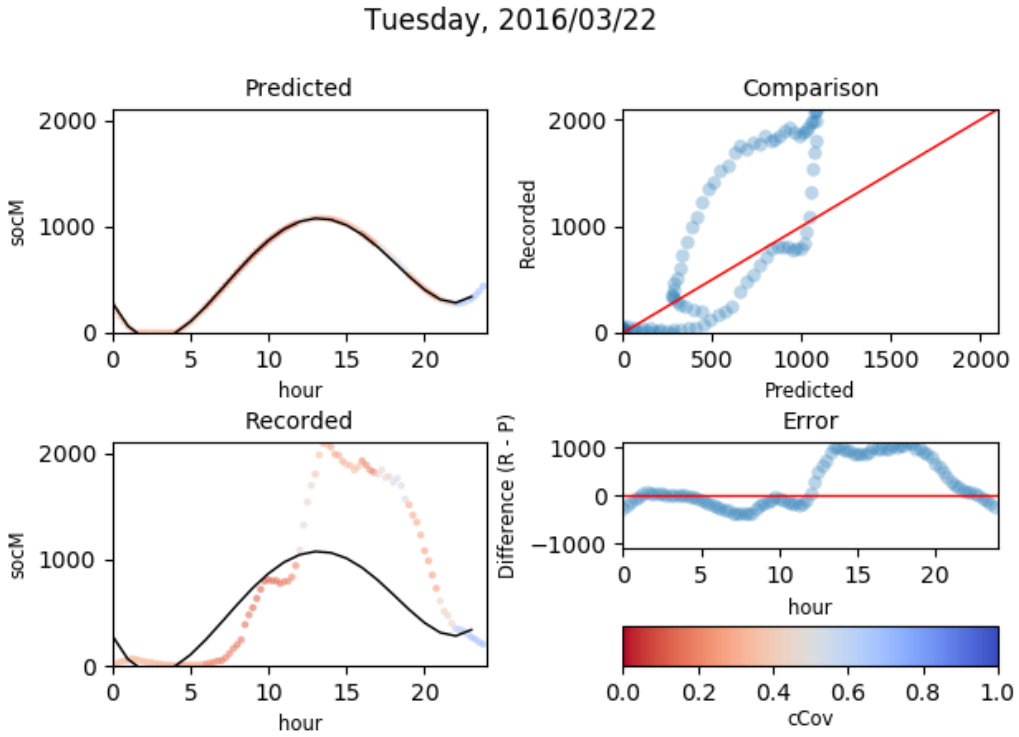


Figure C.45: CS2:QEOP WiFi Forecast Model Validation for 2016-03-22

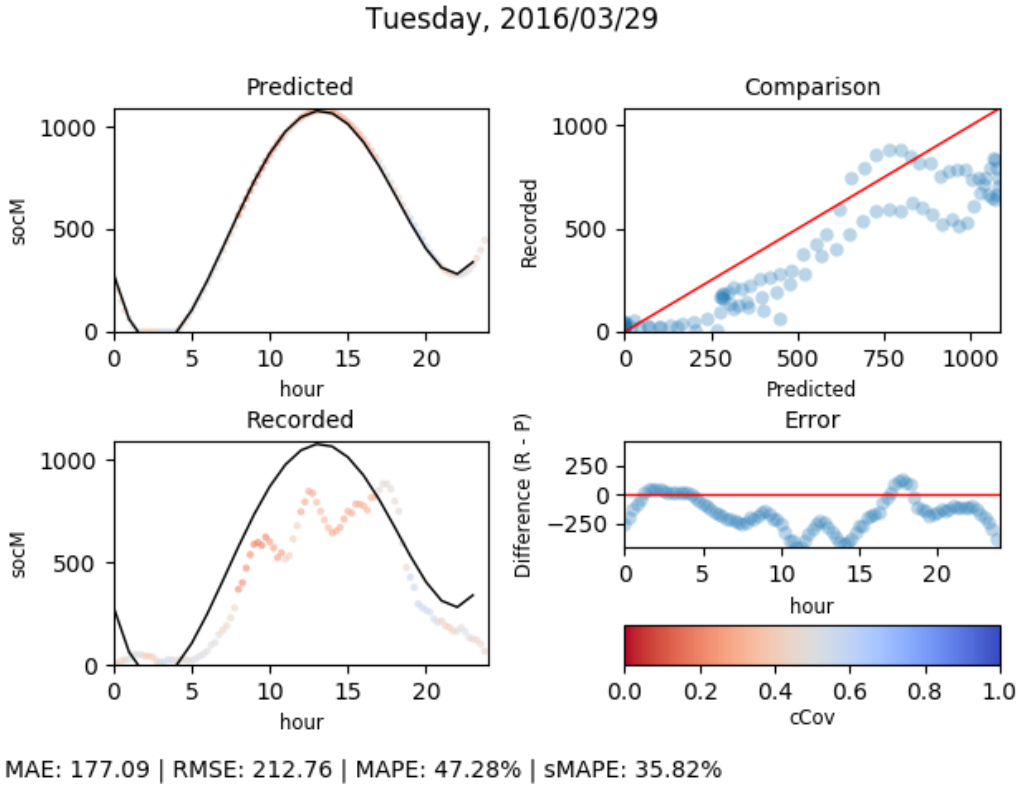


Figure C.46: CS2:QEOP WiFi Forecast Model Validation for 2016-03-29

C.5 CS2:QEOP Forecast Model Validation - WiFi - Naive

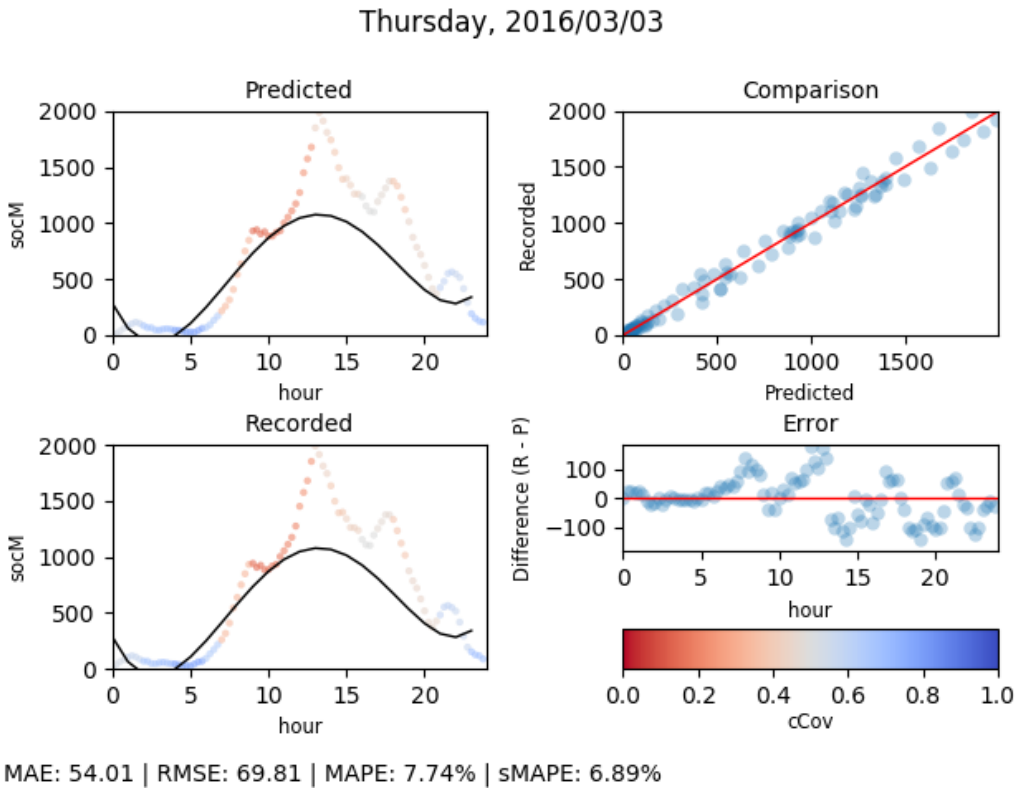


Figure C.47: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-03

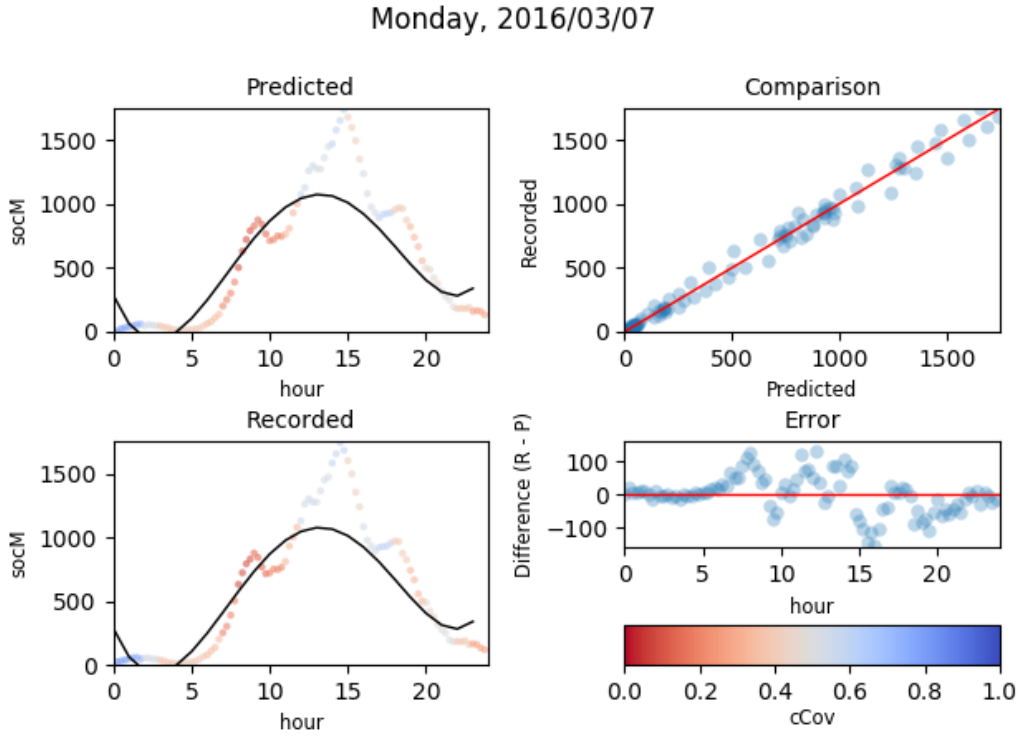


Figure C.48: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-07

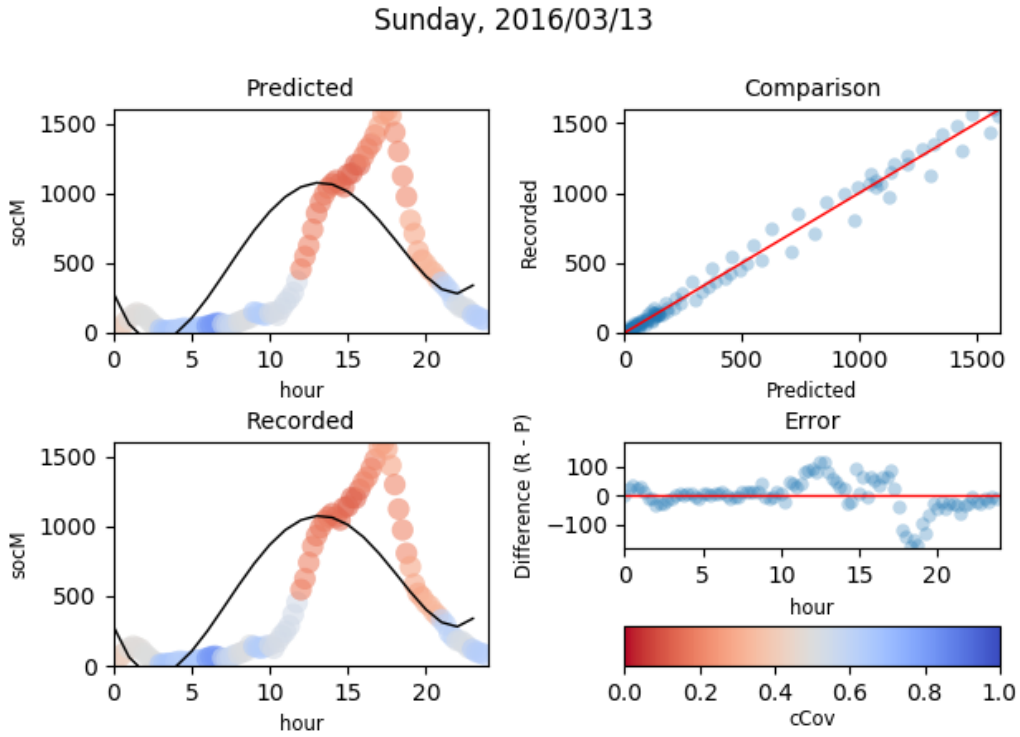
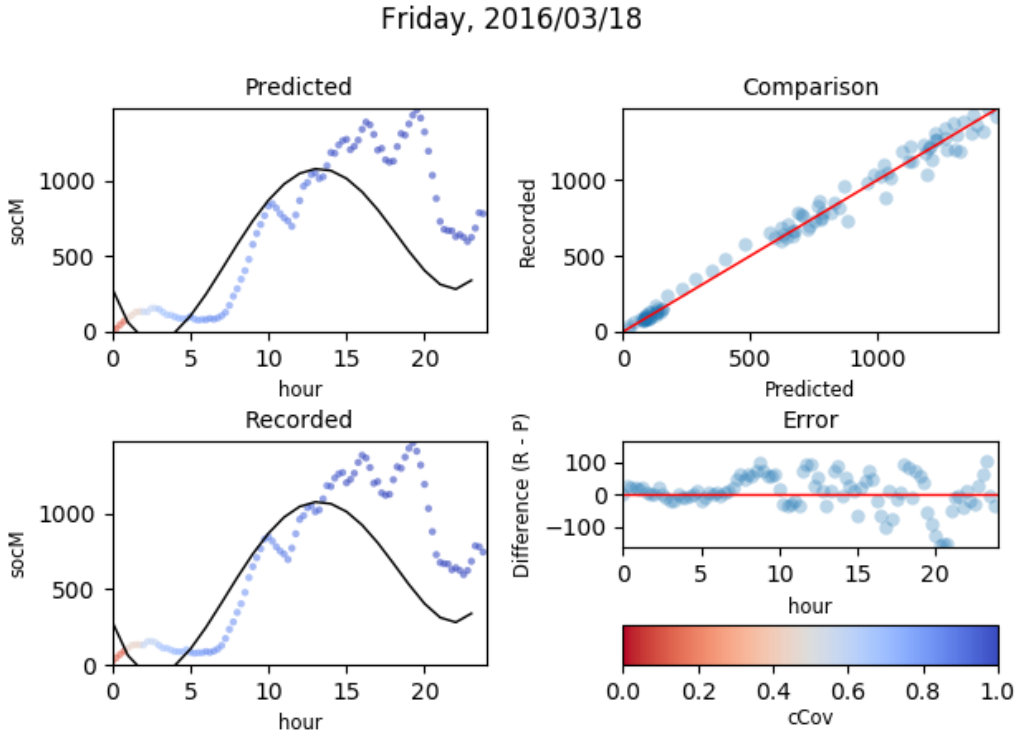
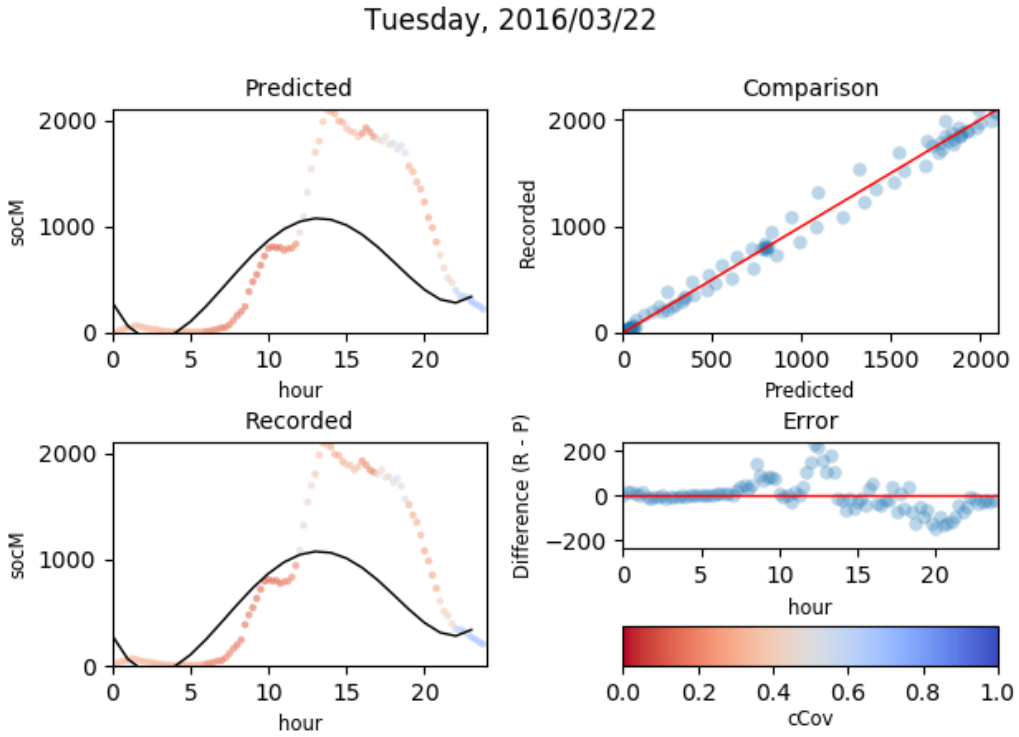


Figure C.49: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-13



MAE: 40.28 | RMSE: 54.60 | MAPE: 5.89% | sMAPE: 4.40%

Figure C.50: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-18



MAE: 47.14 | RMSE: 69.75 | MAPE: 5.99% | sMAPE: 7.66%

Figure C.51: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-22

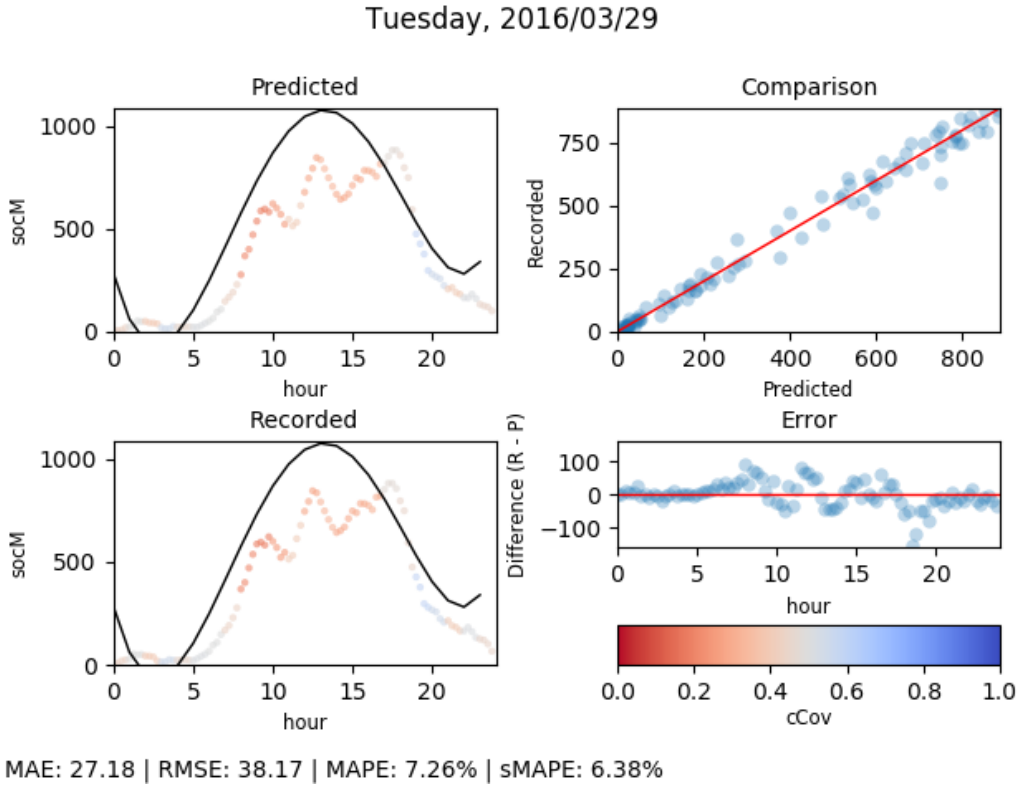


Figure C.52: CS2:QEOP WiFi Naive Forecast Model Validation for 2016-03-29

Acronyms

ABM

Agent-Based Model. 33–35, 66, 69–71, 73, 76–78, 80–86, 89, 90, 92, 93, 118–120, 123, 134, 138–142, 160, 175, 176, 186, 187, 202, 203, 208, 209, 233, 234, 236, 244, 246, 247, 250, 252, 253, 262, 269–271, 273, 274, 280, 282, 283, 286, 289, 290, 292, 297, 299, 304, 305, 307, 308, 313–315, 317–323, 325, 349, 398

AGI

Ambient Geospatial Information. 109, 149

API

Application Programming Interface. 114, 146, 152, 188, 209, 211–216, 219, 227, 254, 255, 293, 295, 333, 336, 337

BD

Big Data. 95, 96, 101, 103–109, 111–113

CA

Cellular Automata. 69, 70, 74, 75, 80, 92, 138, 398

CASA

Centre for Advanced Spatial Analysis. 114, 161, 251

CS1:HyP

Case Study 1: Hyde Park. 150, 207, 244, 245, 249, 250, 252–255, 260, 264–266, 269, 270, 273, 274, 280, 284, 286, 289, 292, 295, 305, 308, 311, 315, 316, 346

CS2:QEOP

Case Study 2: Queen Elizabeth Olympic Park. 146, 150, 249–254, 257, 262, 264, 270, 280, 286, 289, 290, 292, 295, 296, 300, 304, 305, 307, 308, 311, 315, 316, 346

DTM

Digital Terrain Model. 235, 272, 273

GLM

Generalized Linear Model. 264, 265, 295, 296, 308, 319, 323

HyP

Hyde Park. 150, 207–211, 260, 266, 267, 309–311

IBM

Individual-Based Model. 65, 68–70, 92, 99, 138, 314

ICRI

Intel Collaborative Research Institute. 251

IoT

Internet of Things. 251, 321

LLDC

London Legacy Development Corporation. 147, 251

MAE

mean absolute error. 241

MAPE

mean absolute percentage error. 241

MRPE

Mean Relative Percentage Error. 275, 276, 284

MSM

Microsimulation Model. 69, 70, 92, 138, 398

ODD

Overview, Design concepts, and Details. 83, 139, 202, 315, 320

OLS

Ordinary Least Squares. 265

OOP

Object-Oriented Programming. 83, 139–141

OSM

OpenStreetMap. 234, 271–273

PFSM

Probabilistic Finite-State Machine. 183, 202, 351

PSA

Public Space Activity. 208, 209, 233, 246, 249, 250, 252, 253, 269, 290, 292, 294, 295, 298, 307, 308, 310, 313, 317–320, 325, 349

PSU

Public Space Use. 33, 34, 65, 66, 84, 89, 96, 116, 117, 119, 120, 139, 140, 175, 184, 202, 203, 313, 314, 317, 320

QEOP

Queen Elizabeth Olympic Park. 249–255, 257, 260, 264, 266, 267, 269–272, 274, 279–282, 286, 289, 290, 309–311

RMSE

root mean squared error. 241

RT

Real-Time. 208, 209, 246

RTD

Real-Time Data. 33, 34, 92, 95–100, 103, 105, 111–113, 115–118, 120, 125, 128, 137, 203, 207, 208, 250, 291–297, 307, 313–315, 317, 320, 323, 325

RW

Randow Walk Algorithm. 88, 189, 191, 196

SDM

Spatial Disaggregation Model. 133–135, 137–139, 142, 143, 147, 207–209, 233, 241, 246, 249, 250, 253, 262, 273, 280, 282, 283, 289, 307, 315, 323, 356

SF

Social Forces Model. 88, 135

sMAPE

symmetric mean absolute percentage error. 241, 281

SocM

Social Media. 146, 148–151, 153–158, 208, 209, 213, 214, 216–218, 220, 226–230, 240, 241, 250, 253–257, 264–269, 280, 281, 286, 289, 290, 293–297, 316

SPA

Shortest Path Algorithm. 88, 189, 191

TfL

Transport for London. 146

UCL

University College London. 251

VGI

Volunteered Geospatial Information. 109, 270

Glossary

Agent-Based Model

A modelling paradigm focussing on capturing disaggregated dynamic properties of systems, by simulating the individual entities in the system of interest and their interactions. 33, 66, 69–71, 118, 120, 123, 134, 160, 175, 208, 233, 250, 292, 307, 313, 314, 325, 349, 391, 398

Ambient Geospatial Information

A term for geospatial datasets constructed through the collection and extraction of meta-data from other datasets, introduced by Stefanidis et al. (2011). 109, 149, 391

Application Programming Interface

An Application Programming Interface is a particular set of rules and specifications that a software program can follow to access and make use of the services and resources provided by another particular software program that implements that API. 114, 146, 188, 209, 254, 293, 333, 391

Big Data

Datasets too big and complex to be analyzed by conventional data processing tools, often presenting large variety in content an coverage, and produced at rapid speed. 95, 101, 103, 104, 391

Case Study 1: Hyde Park

The first case study undertaken in this work, focussing on Hyde Park. 150, 207, 249, 292, 315, 346, 392

Case Study 2: Queen Elizabeth Olympic Park

The second case study undertaken in this work, focussing on Queen Elizabeth Olympic Park. 146, 249, 292, 315, 346, 392

Cellular Automata

A modelling paradigm able to capture dynamic properties of systems, by simulating the individual entities in the system of interest, often incorporating grid space. 69, 138, 391, 398

Centre for Advanced Spatial Analysis

A research centre at the Bartlett Faculty of the Built Environment, at University College London, focussing on spatial analysis of cities. 114, 161, 251, 392

Digital Terrain Model

A virtual three-dimensional representation of terrain surface. 235, 272, 392

Generalized Linear Model

A statistical linear model. 264, 295, 319, 392

Individual-Based Model

An umbrella term for disaggregated dynamic models that examine a system as a collection of individual entities and their interactions. Includes Cellular Automata (CA), Microsimulation Models (MSMs), and Agent-Based Models (ABMs). 65, 92, 99, 138, 314, 392

Intel Collaborative Research Institute

An industry/academic collaboration between Intel, University College London, Imperial College London and Future Cities Catapult in collaboration with the London Legacy Development Corporation at the Queen Elizabeth Olympic Park. 251, 392

Internet of Things

The networking of physical devices and sensors, enabling the automated collection and exchange of data. 251, 321, 393

London Legacy Development Corporation

The London Legacy Development Corporation is the entity responsible for developing and managing the physical assets (spaces, buildings, etc.) that were built for the London Olympics in 2012. 147, 251, 393

Mean Relative Percentage Error

A method for calculating error between sets of unequal size. 275, 284, 393

Microsimulation Model

A modelling paradigm focussing on capturing disaggregated properties of systems, by simulating the individual entities in the system of interest. 69, 70, 138, 393, 398

navMesh

Short for *Navigation Mesh*, a graph representation of a walkable surface. 189, 191

Object-Oriented Programming

A programming paradigm based on the concept of discrete elements called 'objects', which have control over their self and all data contained in them, and can interact with other objects. 83, 139, 140, 393

Ordinary Least Squares

A method for estimating the unknown parameters in a linear regression model. 265, 393

Overview, Design concepts, and Details

A paradigm for defining and describing an agent-based model, proposed by Grimm et al. (2006, 2010). 83, 139, 202, 315, 393

Probabilistic Finite-State Machine

A mathematical model of computation consisting of defined states and transition rules from one state to another. A system represented as a PFSM can be in exactly one of its defined states at any point in time, and can change state through stochastic (i.e. probabilistically) and/or deterministic (i.e. based on predetermined sequences) processes. 183, 202, 351, 394

Public Space Activity

The act of engaging in an activity in a public space. 208, 233, 249, 292, 307, 308, 313, 325, 349, 394

Public Space Use

The act of engaging in an activity in a public space. 33, 34, 65, 66, 84, 96, 120, 139, 175, 313, 314, 394

Randow Walk Algorithm

A stochastic process that describes a path consisting of a series of random steps in space. 88, 189, 191, 196, 394

Real-Time

Referring to an on-going event or a process which happens instantaneously. 208, 394

Real-Time Data

Any dataset or data point published at the point of capture, therefore referring to an on-going event. 33, 34, 92, 95, 100, 120, 125, 203, 207, 250, 291, 293, 313, 314, 325, 394

Shortest Path Algorithm

An algorithm that provides a solution to the problem of finding an efficient path between two nodes in a graph. Famous implementations include Dijkstra's algorithm and the A* algorithm. 88, 189, 395

Social Forces Model

A model for simulating the movement of pedestrians in crowds and physical environments, incorporating attracting and repelling forces, first proposed by Helbing and Molnár (1995). 88, 135, 395

Social Media

Referring to anything related to online social media - platforms, events, data points, APIs, etc. Often used as shorthand for social media events captured during collection. 146, 148, 154, 208, 213, 240, 250, 254, 280, 293, 316, 395

Spatial Disaggregation Model

A sub-model developed in this work, used for calculating dispersed user activity in public spaces. 133, 134, 139, 142, 147, 207, 249, 280, 282, 307, 315, 356, 395

Transport for London

Metropolitan authority responsible for mass transport systems in the metropolitan London area. 146, 395

University College London

A teaching and research academic institution in London, UK. 251, 395

Volunteered Geospatial Information

A term for geospatial datasets constructed by members of the public through active participation, introduced by Goodchild (2007). 109, 270, 395

Bibliography

Aly, M. (2008). Real time detection of lane markers in urban streets. In *2008 IEEE Intelligent Vehicles Symposium*, pages 7–12.

Anderson, C. (2008). The End of Theory: The Data Deluge Makes the Scientific Method Obsolete.

Angus, S. D. and Hassani-Mahmooei, B. (2015). "Anarchy" Reigns: A Quantitative Analysis of Agent-Based Modelling Publication Practices in JASSS, 2001-2012. *Journal of Artificial Societies and Social Simulation*, 18(4):16.

Appleyard, D. and Lintell, M. (1972). The Environmental Quality of City Streets: The Residents' Viewpoint. *Journal of the American Institute of Planners*, 38(2):84–101.

Aspinall, P., Mavros, P., Coyne, R., and Roe, J. (2013). The urban brain: analysing outdoor physical activity with mobile EEG. *British Journal of Sports Medicine*, pages bjsports–2012–091877.

Avvenuti, M., Cresci, S., Marchetti, A., Meletti, C., and Tesconi, M. (2016). Predictability or Early Warning: Using Social Media in Modern Emergency Response. *IEEE Internet Computing*, 20(6):4–6.

Bandini, S., Crociani, L., and Vizzari, G. (2014a). Heterogeneous Speed Profiles in Discrete Models for Pedestrian Simulation. *arXiv:1401.8132 [cs]*. arXiv: 1401.8132.

Bandini, S., Gorrini, A., and Vizzari, G. (2014b). Towards an Integrated Approach to Crowd Analysis and Crowd Synthesis: a Case Study and First Results. *Pattern Recognition Letters*, 44:16–29. arXiv: 1303.5029.

Banerjee, T. (2001). The Future of Public Space: Beyond Invented Streets and Reinvented Places. *Journal of the American Planning Association*, 67(1):9–24.

Barth, D. (2009). The bright side of sitting in traffic: Crowdsourcing road congestion data.

Batty, M. (2001). Models in planning: technological imperatives and changing roles. *International Journal of Applied Earth Observation and Geoinformation*, 3(3):252–266.

Batty, M. (2005). *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. The MIT Press.

Batty, M. (2008). Fifty Years of Urban Modeling: Macro-Statics to Micro-Dynamics. In Albeverio, P. D. D. h. c. S., Andrey, D., Giordano, P., and Vancheri, D. A., editors, *The Dynamics of Complex Urban Systems*, pages 1–20. Physica-Verlag HD. DOI: 10.1007/978-3-7908-1937-3_1.

Batty, M. (2012). A Generic Framework for Computational Spatial Modelling. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 19–50. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_2.

Batty, M., Axhausen, K. W., Giannotti, F., Pozdnoukhov, A., Bazzani, A., Wachowicz, M., Ouzounis, G., and Portugali, Y. (2012). Smart cities of the future. *The European Physical Journal Special Topics*, 214(1):481–518.

Batty, M., Desyllas, J., and Duxbury, E. (2003). Safety in Numbers? Modelling Crowds and Designing Control for the Notting Hill Carnival. *Urban Studies*, 40(8):1573–1590.

Batty, M. and Hudson-Smith, A. (2005). Urban Simulacra: London. *Architectural Design*, 75(6):42–47.

Batty, M., Hudson-Smith, A., Milton, R., and Crooks, A. (2010). Map mashups, Web 2.0 and the GIS revolution. *Annals of GIS*, 16(1):1–13.

Becker, H., Naaman, M., and Gravano, L. (2011). Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM*, 11:438–441.

Benenson, I. and Torrens, P. M. (2004). *Geosimulation: Automata-based Modeling of Urban Phenomena*. John Wiley & Sons. Google-Books-ID: 8hPsq78L5qcC.

Berling-Wolff, S. and Wu, J. (2004). Modeling urban landscape dynamics: A review. *Ecological Research*, 19(1):119–129.

Birkin, M. and Wu, B. (2012). A Review of Microsimulation and Hybrid Agent-Based Approaches. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 51–68. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_3.

Bitgood, S. and Dukes, S. (2006). Not Another Step! Economy of Movement and Pedestrian Choice Point Behavior in Shopping Malls. *Environment and Behavior*, 38(3):394–405.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99(90003):7280–7287.

Boot, J. C. G., Feibes, W., and Lisman, J. H. C. (1967). Further Methods of Derivation of Quarterly Figures from Annual Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 16(1):65–75.

boyd, d. and Crawford, K. (2012). Critical Questions for Big Data. *Information, Communication & Society*, 15(5):662–679.

Broad, E. (2015). Closed, shared, open data: whats in a name?

Calabrese, F. (2009). WikiCity: Real-Time Location-Sensitive Tools for the City. In Foth, M., editor, *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, pages 390–413. IGI Global, Hershey, PA, USA.

Calabrese, F., Colonna, M., Lovisolo, P., Parata, D., and Ratti, C. (2011). Real-Time Urban Monitoring Using Cell Phones: A Case Study in Rome. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):141–151.

Canter, D. and Tagg, S. K. (1975). Distance Estimation in Cities. *Environment and Behavior*, 7(1):59–80.

Carmona, M. (2010a). Contemporary Public Space: Critique and Classification, Part One: Critique. *Journal of Urban Design*, 15(1):123–148.

Carmona, M. (2010b). Contemporary Public Space, Part Two: Classification. *Journal of Urban Design*, 15(2):157–173.

Carr, S. (1992). *Public Space*. Cambridge University Press.

Cassa, C. A., Chunara, R., Mandl, K., and Brownstein, J. S. (2013). Twitter as a Sentinel in Emergency Situations: Lessons from the Boston Marathon Explosions. *PLOS Currents Disasters*.

Castle, C. J. E., Waterson, N. P., Pellissier, E., and Bail, S. L. (2011). A Comparison of Grid-based and Continuous Space Pedestrian Modelling Software: Analysis of Two UK Train Stations. In Peacock, R. D., Kuligowski, E. D., and Averill, J. D., editors, *Pedestrian and Evacuation Dynamics*, pages 433–446. Springer US.

Cheliotis, K. (2016). Capturing Real-Time Public Space Activity Using Publicly Available Digital Traces. In *International AAAI Conference on Web and Social Media; Tenth International AAAI Conference on Web and Social Media*.

Chiesura, A. (2004). The role of urban parks for the sustainable city. *Landscape and Urban Planning*, 68(1):129–138.

Ciolek, D. T. M. (1983). The proxemics lexicon: A first approximation. *Journal of Nonverbal Behavior*, 8(1):55–79.

Ciolek, M. (1976). Location of static gatherings in pedestrian areas: an exploratory study. In *Ergonomics Society of Australia and New Zealand Conference, 13th, 1976, Canberra*.

Ciolek, T. M. and Kendon, A. (1980). Environment and the Spatial Arrangement of Conversational Encounters. *Sociological Inquiry*, 50(3-4):237–271.

Cohen, R. and Ruths, D. (2013). Classifying Political Orientation on Twitter: Its Not Easy! *International AAAI Conference on Web and Social Media; Seventh International AAAI Conference on Weblogs and Social Media*.

Conroy-Dalton, R. (2003). The Secret Is To Follow Your Nose Route Path Selection and Angularity. *Environment and Behavior*, 35(1):107–131.

Costa, M. (2010). Interpersonal Distances in Group Walking. *Journal of Nonverbal Behavior*, 34(1):15–26.

Crociani, L., Yanagisawa, D., Vizzari, G., Nishinari, K., and Bandini, S. (2016). Avoid or Follow? Modelling Route Choice Based on Experimental Empirical Evidences. *arXiv:1610.07901 [cs]*. arXiv: 1610.07901.

Crooks, A., Croitoru, A., Lu, X., Wise, S., Irvine, J. M., and Stefanidis, A. (2015). Walk This Way: Improving Pedestrian Agent-Based Models through Scene Activity Analysis. *ISPRS International Journal of Geo-Information*, 4(3):1627–1656.

Crooks, A., Heppenstall, A., and Malleson, N. (2018). Agent-Based Modeling. In Huang, B., editor, *Comprehensive Geographic Information Systems*, pages 218–243. Elsevier, Oxford.

Crooks, A. T. and Heppenstall, A. J. (2012). Introduction to Agent-Based Modelling. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors,

Agent-Based Models of Geographical Systems, pages 85–105. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_5.

Dai, J., Li, X., and Liu, L. (2013). Simulation of pedestrian counter flow through bottlenecks by using an agent-based model. *Physica A: Statistical Mechanics and its Applications*, 392(9):2202–2211.

D'Andrea, E., Ducange, P., Lazzerini, B., and Marcelloni, F. (2015). Real-Time Detection of Traffic From Twitter Stream Analysis. *IEEE Transactions on Intelligent Transportation Systems*, 16(4):2269–2283.

Dawkins, O., Dennett, A., and Hudson-Smith, A. (2018). Living with a Digital Twin: Operational management and engagement using IoT and Mixed Realities at UCL's Here East Campus on the Queen Elizabeth Olympic Park.

Dia, H. (2002). An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transportation Research Part C: Emerging Technologies*, 10(56):331–349.

Dias, C., Ejtemai, O., Sarvi, M., and Burd, M. (2014). Exploring Pedestrian Walking through Angled Corridors. *Transportation Research Procedia*, 2:19–25.

Durupinar, F., Pelechano, N., Allbeck, J. M., Gdkbay, U., and Badler, N. I. (2011). How the Ocean Personality Model Affects the Perception of Crowds. *IEEE Computer Graphics and Applications*, 31(3):22–31.

Dutton, W. H., Blumler, J. G., and Kraemer, K. L., editors (1987). *Wired Cities: Shaping the Future of Communications*. G. K. Hall & Co., Boston, MA, USA.

Easterling, D. R., Horton, B., Jones, P. D., Peterson, T. C., Karl, T. R., Parker, D. E., Salinger, M. J., Razuvayev, V., Plummer, N., Jamason, P., and Folland, C. K. (1997). Maximum and Minimum Temperature Trends for the Globe. *Science*, 277(5324):364–367.

Edmonds, B. and Moss, S. (2005). From KISS to KIDS An Anti-simplistic Modelling Approach. In Davidsson, P., Logan, B., and Takadama, K., editors, *Multi-Agent and Multi-Agent-Based Simulation*, number 3415 in Lecture Notes in Computer Science, pages 130–144. Springer Berlin Heidelberg.

Erkip, F. (2003). The Shopping Mall as an Emergent Public Space in Turkey. *Environment and Planning A: Economy and Space*, 35(6):1073–1093.

Fang, J., El-Tawil, S., and Aguirre, B. (2016). Leaderfollower model for agent based simulation of social collective behavior during egress. *Safety Science*, 83:40–47.

Foth, M., Hudson-Smith, A., and Gifford, D. (2016). Smart cities, social capital, and citizens at play: A critique and a way forward. In Olleros, F. X. and Zhegu, M., editors, *Research Handbook on Digital Transformations*, pages 203–221. Edward Elgar Publishing, Cheltenham, United Kingdom.

Gärling, T. and Gärling, E. (1988). Distance Minimization in Downtown Pedestrian Shopping. *Environment and Planning A*, 20(4):547–554.

Gehl, J. (1987). *Life between buildings: using public space*. Van Nostrand Reinhold, New York.

Gehl, J. and Gemzøe, L. (2000). *New city spaces*. Danish Architectural Press.

Gehl Architects (2004). *Towards a Fine City for People, Public Space, and Public Life - London 2004*. Gehl Architects, Copenhagen.

Geiger, A., Ziegler, J., and Stiller, C. (2011). StereoScan: Dense 3d reconstruction in real-time. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968.

Golledge, R., Jacobson, R. D., Kitchin, R., and Blades, M. (2000). Cognitive Maps, Spatial Abilities, and Human Wayfinding. *Geographical review of Japan, Series B.*, 73(2):93–104.

Golledge, R. G. (1995). Path selection and route preference in human navigation: A progress report. In Frank, A. U. and Kuhn, W., editors, *Spatial Information*

Theory A Theoretical Basis for GIS: International Conference COSIT '95 Semmering, Austria, September 2123, 1995 Proceedings, pages 207–222. Springer Berlin Heidelberg, Berlin, Heidelberg.

Golledge, R. G. (1999). *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes*. JHU Press. Google-Books-ID: TjzxpAWiamUC.

Goodchild, M. F. (2007). Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69(4):211–221.

Graham, S. (1997). Cities in the Real-Time Age: The Paradigm Challenge of Telecommunications to the Conception and Planning of Urban Space. *Environment and Planning A*, 29(1):105–127.

Graham, S. and Marvin, S. (1999). Planning cybercities: integrating telecommunications into urban planning. *Town Planning Review*, 70(1):89.

Gray, S., Milton, R., and Hudson-Smith, A. (2015). Advances in Crowdsourcing: Surveys, Social Media and Geospatial Analysis: Towards a Big Data Toolkit. In Garrigos-Simon, F. J., Gil-Pechun, I., and Estelles-Miguel, S., editors, *Advances in Crowdsourcing*, pages 163–179. Springer International Publishing. DOI: 10.1007/978-3-319-18341-1_13.

Gray, S., O'Brien, O., and Hügel, S. (2016). Collecting and Visualizing Real-Time Urban Data through City Dashboards. *Built Environment*, 42(3):498–509.

Greenwood, S., Perrin, r., and Duggan, M. (2016). Social Media Update 2016. Technical report, Pew Research Centre.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jrgensen, C., Mooij, W. M., Mller, B., Peer, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Rger, N., Strand, E., Souissi, S., Stillman, R. A., Vab, R., Visser, U., and DeAngelis, D. L. (2006). A standard protocol

for describing individual-based and agent-based models. *Ecological Modelling*, 198(12):115–126.

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010). The ODD protocol: A review and first update. *Ecological Modelling*, 221(23):2760–2768.

Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

Guerrero, V. M. (1990). Temporal Disaggregation of Time Series: An ARIMA-Based Approach. *International Statistical Review / Revue Internationale de Statistique*, 58(1):29–46.

Guy, S. J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., and Manocha, D. (2010). PLEdestrians: A Least-effort Approach to Crowd Simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '10, pages 119–128, Goslar Germany, Germany. Eurographics Association.

Haase, D. and Schwarz, N. (2009). Simulation Models on Human–Nature Interactions in Urban Landscapes: A Review Including Spatial Economics, System Dynamics, Cellular Automata and Agent-based Approaches. *Living Rev. Landscape Res.*, 3.

Haciomeroglu, M., Laycock, R. G., and Day, A. M. (2008). Dynamically populating large urban environments with ambient virtual humans. *Computer Animation and Virtual Worlds*, 19(3-4):307–317.

Hall, E. T. (1963). A System for the Notation of Proxemic Behavior. *American Anthropologist*, 65(5):1003–1026.

Hall, E. T. (1966a). *The hidden dimension*. Doubleday, Garden City, N.Y. OCLC: 203769.

Hall, P. G. (1966b). *Von Thunen's isolated state : an English edition of Der isolierte Staat*. Oxford ; London : Pergamon Press, 1st ed edition.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107.

Hartmann, D. and Hasel, P. (2014). Efficient Dynamic Floor Field Methods for Microscopic Pedestrian Crowd Simulations. *Communications in Computational Physics*, 16(1):264–286.

Harvey, D. (2013). *Rebel Cities: From the Right to the City to the Urban Revolution*. Verso Books, London, 2 edition edition.

Heath, B., Hill, R., and Ciarallo, F. (2009). A Survey of Agent-Based Modeling Practices (January 1998 to July 2008). *Journal of Artificial Societies and Social Simulation*, 12(4):9.

Helbing, D., Buzna, L., Johansson, A., and Werner, T. (2005). Self-Organized Pedestrian Crowd Dynamics: Experiments, Simulations, and Design Solutions. *Transportation Science*, 39(1):1–24.

Helbing, D. and Johansson, A. (2011). Pedestrian, Crowd and Evacuation Dynamics. In Ph.D, R. A. M., editor, *Extreme Environmental Events*, pages 697–716. Springer New York. DOI: 10.1007/978-1-4419-7695-6_37.

Helbing, D. and Molnár, P. (1995). Social force model for pedestrian dynamics. *Physical Review E*, 51(5):4282–4286.

Heppenstall, A., Malleson, N., and Crooks, A. (2016). Space, the Final Frontier: How Good are Agent-Based Models at Simulating Individuals and Space in Cities? *Systems*, 4(1):9.

Hillier, B. and Iida, S. (2005). Network and Psychological Effects in Urban Movement. In Cohn, A. G. and Mark, D. M., editors, *Spatial Information Theory:*

International Conference, COSIT 2005, Ellicottville, NY, USA, September 14-18, 2005. Proceedings, pages 475–490. Springer Berlin Heidelberg, Berlin, Heidelberg.

Holcombe, M., Adra, S., Bicak, M., Chin, S., Coakley, S., Graham, A. I., Green, J., Greenough, C., Jackson, D., Kiran, M., MacNeil, S., Maleki-Dizaji, A., McMinn, P., Pogson, M., Poole, R., Qwarnstrom, E., Ratnieks, F., Rolfe, M. D., Smallwood, R., Sun, T., and Worth, D. (2012). Modelling complex biological systems using an agent-based approach. *Integrative Biology*, 4(1):53.

Hollands, R. G. (2008). Will the real smart city please stand up? *City*, 12(3):303–320.

Hudson-Smith, A. (2014). Tracking, Tagging and Scanning the City. *Architectural Design*, 84(1):40–47.

Hudson-Smith, A., Crooks, A., Gibin, M., Milton, R., and Batty, M. (2009). Neogeography and Web 2.0: concepts, tools and applications. *Journal of Location Based Services*, 3(2):118–145.

Iacono, M., Levinson, D., and El-Geneidy, A. (2008). Models of Transportation and Land Use Change: A Guide to the Territory. *Journal of Planning Literature*, 22(4):323–340.

Iltanen, S. (2012). Cellular Automata in Urban Spatial Modelling. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 69–84. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_4.

Ipsos Mori (2015a). Royal Parks Stakeholder Research Programme 2014 - Park profile: Hyde Park (Waves 1-3). Technical report, The Royal Parks.

Ipsos Mori (2015b). Royal Parks Stakeholder Research Programme 2014 - Park visitors research. Technical report, The Royal Parks.

Ishaque, M. M. and Noland, R. B. (2008). Behavioural Issues in Pedestrian Speed Choice and Street Crossing Behaviour: A Review. *Transport Reviews*, 28(1):61–85.

Ishida, T. and Isbister, K. (2000). *Digital Cities: Technologies, Experiences, and Future Perspectives*. Springer Science & Business Media. Google-Books-ID: c8N89LbQsTQC.

Jacobs, J. (1961). *The Death and Life of Great American Cities*. Knopf Doubleday Publishing Group.

Jazwinski, C. H. and Walcheski, C. H. (2011). At the Mall With Children: Group Size and Pedestrian Economy of Movement. *Environment and Behavior*, 43(3):363–386.

Jongman, B., Wagemaker, J., Romero, B. R., and de Perez, E. C. (2015). Early Flood Detection for Rapid Humanitarian Response: Harnessing Near Real-Time Satellite and Twitter Signals. *ISPRS International Journal of Geo-Information*, 4(4):2246–2266.

Karlenzig, W., Marquardt, F., and Yaseen, R. (2007). *How Green Is Your City?: The SustainLane US City Rankings*. New Society Publishers, Gabriola Island, B.C.

Kay, A. C. (1993). The Early History of Smalltalk. In *The Second ACM SIGPLAN Conference on History of Programming Languages*, HOPL-II, pages 69–95, New York, NY, USA. ACM.

Kitchin, R. (2013). The real-time city? Big data and smart urbanism. *GeoJournal*, 79(1):1–14.

Kitchin, R. (2014). *The Data Revolution: Big Data, Open Data, Data Infrastructures and Their Consequences*. SAGE Publications Ltd, London.

Kitchin, R., Lauriault, T. P., and McArdle, G. (2015). Knowing and governing cities through urban indicators, city benchmarking and real-time dashboards. *Regional Studies, Regional Science*, 2(1):6–28.

Kitchin, R. and McArdle, G. (2016). What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. *Big Data & Society*, 3(1):2053951716631130.

Kokkinogenis, Z., Filguieras, J., Carvalho, S., Sarmento, L., and Rossetti, R. J. F. (2015). Mobility Network Evaluation in the User Perspective: Real-Time Sensing of Traffic Information in Twitter Messages. In *Advances in Artificial Transportation Systems and Simulation*, pages 219–234. Academic Press, Boston. DOI: 10.1016/B978-0-12-397041-1.00012-1.

Komninos, N. (2013). *Intelligent Cities: Innovation, Knowledge Systems and Digital Spaces*. Routledge. Google-Books-ID: ibcCsT3474cC.

Kouvelas, A., Aboudolas, K., Papageorgiou, M., and Kosmatopoulos, E. B. (2011). A Hybrid Strategy for Real-Time Traffic Signal Control of Urban Road Networks. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):884–894.

Lamarche, F. and Donikian, S. (2004). Crowd of Virtual Humans: a New Approach for Real Time Navigation in Complex and Structured Environments. *Computer Graphics Forum*, 23(3):509–518.

Larco, N. (2003). What is Urban? *Places*, 15(2).

Lee, W.-H., Tseng, S.-S., and Tsai, S.-H. (2009). A knowledge based real-time travel time prediction system for urban network. *Expert Systems with Applications*, 36(3, Part 1):4239–4247.

Leng, B., Wang, J., and Xiong, Z. (2015). Pedestrian simulations in hexagonal cell local field model. *Physica A: Statistical Mechanics and its Applications*, 438:532–543.

Lin, Y.-H., Liu, Y.-S., Gao, G., Han, X.-G., Lai, C.-Y., and Gu, M. (2013). The IFC-based path planning for 3d indoor spaces. *Advanced Engineering Informatics*, 27(2):189–205.

Lisman, J. H. C. and Sandee, J. (1964). Derivation of Quarterly Figures from Annual Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 13(2):87–90.

Liu, S., Lo, S., Ma, J., and Wang, W. (2014). An Agent-Based Microscopic Pedestrian Flow Simulation Model for Pedestrian Traffic Problems. *IEEE Transactions on Intelligent Transportation Systems*, 15(3):992–1001.

Loukaitou-Sideris, A. (1996). Cracks in the city: Addressing the constraints and potentials of urban design. *Journal of Urban Design*, 1(1):91–103.

Lowe, M. (2005). The Regional Shopping Centre in the Inner City: A Study of Retail-led Urban Regeneration. *Urban Studies*, 42(3):449–470.

Lynch, K. (1960). *The Image of the City*. MIT Press. Google-Books-ID: _phRPWsSpAgC.

Madanipour, A. (1997). Ambiguities of urban design. *Town Planning Review*, 68(3):363.

Malleson, N. (2012). Using Agent-Based Models to Simulate Crime. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 411–434. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_19.

Malleson, N., Evans, A., Heppenstall, A., and See, L. (2010). Evaluating an Agent-Based Model of Burglary. Working Paper 10/01, University of Leeds, School of Geography, Leeds, UK.

Malleson, N., Heppenstall, A., See, L., and Evans, A. (2013). Using an Agent-Based Crime Simulation to Predict the Effects of Urban Regeneration on Individual Household Burglary Risk. *Environment and Planning B: Planning and Design*, 40(3):405–426.

Manley, E. (2013). *Modelling Driver Behaviour to Predict Urban Road Traffic Dynamics*. PhD thesis, University College London, United Kingdom.

Marshall, S. (2005). *Streets and Patterns*. Routledge. Google-Books-ID: 6rBFKX3MmcgC.

Marshall, S. (2012). Science, pseudo-science and urban design. *URBAN DESIGN International*, 17(4):257–271.

Mattern, S. (2015). Mission Control: A History of the Urban Dashboard. *Places Journal*.

Mayer-Schönberger, V. and Cukier, K. (2013). *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. Houghton Mifflin Harcourt.

McArdle, G. and Kitchin, R. (2015). Improving the Veracity of Open and Real-Time Urban Data. SSRN Scholarly Paper ID 2643430, Social Science Research Network, Rochester, NY.

Miller, C., Ginnis, S., Stobart, R., Krasodomski-Jones, A., and Clemence, M. (2015). The road to representivity: a Demos and Ipsos MORI report on sociological research using Twitter. Technical report, Demos, London.

Mills, A., Chen, R., Lee, J., and Rao, H. R. (2009). Web 2.0 Emergency Applications: How Useful Can Twitter be for Emergency Response? *Journal of Information Privacy and Security*, 5(3):3–26.

Min, W. and Wynter, L. (2011). Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4):606–616.

Montello, D. R. (1993). Scale and multiple psychologies of space. In Frank, A. U. and Campari, I., editors, *Spatial Information Theory A Theoretical Basis for GIS: European Conference, COSIT'93 Marciana Marina, Elba Island, Italy September 1922, 1993 Proceedings*, pages 312–321. Springer Berlin Heidelberg, Berlin, Heidelberg.

Moussad, M., Helbing, D., and Theraulaz, G. (2011). How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences*, 108(17):6884–6888.

Narang, S., Best, A., Curtis, S., and Manocha, D. (2015). Generating Pedestrian Trajectories Consistent with the Fundamental Diagram Based on Physiological and Psychological Factors. *PLOS ONE*, 10(4):e0117856.

Navarro, L., Flacher, F., and Corruble, V. (2011). Dynamic Level of Detail for Large Scale Agent-based Urban Simulations. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, pages 701–708, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Németh, J. (2006). Conflict, Exclusion, Relocation: Skateboarding and Public Space. *Journal of Urban Design*, 11(3):297–318.

Nikolai, C. and Madey, G. (2009). Tools of the trade: A survey of various agent based modeling platforms. *Journal of Artificial Societies and Social Simulation*, 12(2):2.

Ondej, J., Pettr, J., Olivier, A.-H., and Donikian, S. (2010). A Synthetic-vision Based Steering Approach for Crowd Simulation. In *ACM SIGGRAPH 2010 Papers*, SIGGRAPH '10, pages 123:1–123:9, New York, NY, USA. ACM.

Paddison, R. (1993). City Marketing, Image Reconstruction and Urban Regeneration. *Urban Studies*, 30(2):339–349.

Pailhous, J. (1970). *La reprsentation de l'espace urbain: l'exemple du chauffeur de taxi*, volume 10. Presses universitaires de France.

Pailhous, J. (1984). The representation of urban space: its development and its role in the organisation of journeys. *Social representations*, pages 311–327.

Papadimitriou, E., Yannis, G., and Golias, J. (2009). A critical assessment of pedestrian behaviour models. *Transportation Research Part F: Traffic Psychology and Behaviour*, 12(3):242–255.

Parker, J. and Epstein, J. M. (2011). A Distributed Platform for Global-Scale Agent-Based Models of Disease Transmission. *ACM transactions on modeling and computer simulation : a publication of the Association for Computing Machinery*, 22(1):2.

Pelechano, N., Allbeck, J. M., and Badler, N. I. (2007). Controlling individual agents in high-density crowd simulation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '07, pages 99–108, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Pelechano, N., Allbeck, J. M., and Badler, N. I. (2008). Virtual Crowds: Methods, Simulation, and Control. *Synthesis Lectures on Computer Graphics and Animation*, 3(1):1–176.

Pelechano, N. and Badler, N. I. (2006). Modeling crowd and trained leader behavior during building evacuation. *IEEE computer graphics and applications*, 26(6):80–86.

Penn, A. and Turner, A. (2001). Space syntax based agent simulation. In *Presented at: 1st International Conference on Pedestrian and Evacuation Dynamics, University of Duisburg, Germany. (2001)*, University of Duisburg, Germany.

Perez, T. and Rushing, R. (2007). The CitiStat model: How data-driven government can increase efficiency and effectiveness. Technical report.

Pettre, J., Laumond, J.-P., and Thalmann, D. (2005). A navigation graph for real-time crowd animation on multilayered and uneven terrain. In *First International Workshop on Crowd Simulation*, volume 43, page 194. New York: Pergamon Press.

Pollefeys, M., Nistr, D., Frahm, J.-M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.-J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewnius, H., Yang, R., Welch, G., and Towles, H. (2008). Detailed Real-Time Urban 3d Reconstruction from Video. *International Journal of Computer Vision*, 78(2-3):143–167.

Pouke, M., Goncalves, J., Ferreira, D., and Kostakos, V. (2016). Practical simulation of virtual crowds using points of interest. *Computers, Environment and Urban Systems*, 57:118–129.

Project for Public Spaces (2000). *How to Turn a Place Around: A Handbook for Creating Successful Public Spaces*. Project for Public Spaces, New York, NY.

Railsback, S. F., Lytinen, S. L., and Jackson, S. K. (2006). Agent-based Simulation Platforms: Review and Development Recommendations. *SIMULATION*, 82(9):609–623.

Ratti, C. and Claudel, M. (2016). *The City of Tomorrow: Sensors, Networks, Hackers, and the Future of Urban Life*. Yale University Press.

Reynolds, C. (2006). Big Fast Crowds on PS3. In *Proceedings of the 2006 ACM SIGGRAPH Symposium on Videogames*, Sandbox '06, pages 113–121, New York, NY, USA. ACM.

Reynolds, C. W. (1987). Flocks, Herds and Schools: A Distributed Behavioral Model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA. ACM.

Riddell, R., editor (2004). *Sustainable Urban Planning*. Blackwell Publishing Ltd, Oxford, UK. DOI: 10.1002/9780470773703.

Roberts, P., Sykes, H., and Granger, R. (2016). *Urban Regeneration*. SAGE. Google-Books-ID: tjYEDQAAQBAJ.

Roozemond, D. A. (2001). Using intelligent agents for pro-active, real-time urban intersection control. *European Journal of Operational Research*, 131(2):293–301.

Rowley, A. (1994). Definitions of urban design: The nature and concerns of urban design. *Planning Practice & Research*, 9(3):179–197.

Sailer, K., Koutsolampros, P., Austwick, M. Z., Varoudis, T., and Hudson-Smith, A. (2016). Measuring Interaction in Workplaces. In *Architecture and Interaction*, HumanComputer Interaction Series, pages 137–161. Springer, Cham.

Sailer, K. and Psathiti, C. (2017). A Prospect-Refuge Approach to Seat Preference: Environmental psychology and spatial layout. In *Proceedings of the 11th International Space Syntax Symposium*, volume 11, pages 137.1–137.16. Instituto Superior Tecnico, Departamentode Engenharia Civil, Arquitetura e Georrecursos, Portugal.

Sakaki, T., Okazaki, M., and Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 851–860, New York, NY, USA. ACM.

Sevtsuk, A. (2009). Mapping the MIT Campus in Real Time Using WiFi. In Foth, M., editor, *Handbook of Research on Urban Informatics: The Practice and Promise of the Real-Time City*, pages 326–338. IGI Global, Hershey, PA, USA.

Shamir, U. and Salomons, E. (2008). Optimal Real-Time Operation of Urban Water Distribution Systems Using Reduced Models. *Journal of Water Resources Planning and Management*, 134(2):181–185.

Shao, W. and Terzopoulos, D. (2005). Autonomous Pedestrians. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '05, pages 19–28, New York, NY, USA. ACM.

Shepard, M. (2011). *Sentient City: Ubiquitous Computing, Architecture, and the Future of Urban Space*. The MIT Press.

Shi, W. and Liu, Y. (2010). Real-time urban traffic monitoring with global positioning system-equipped vehicles. *IET Intelligent Transport Systems*, 4(2):113–120.

Singer, N. (2012). I.B.M. Takes Smarter Cities Concept to Rio de Janeiro. *The New York Times*.

Song, X., Ma, L., Ma, Y., Yang, C., and Ji, H. (2016). Selfishness- and Selflessness-based models of pedestrian room evacuation. *Physica A: Statistical Mechanics and its Applications*, 447:455–466.

Southworth, M. and Ben-Joseph, E. (2003). Streets and the Shaping of Towns and Cities. *Bibliovault OAI Repository, the University of Chicago Press*.

Spiers, H. J. and Maguire, E. A. (2008). The dynamic nature of cognition during wayfinding. *Journal of Environmental Psychology*, 28(3):232–249.

Sprague, N., Ballard, D., and Robinson, A. (2007). Modeling Embodied Visual Behaviors. *ACM Trans. Appl. Percept.*, 4(2).

Stefanidis, A., Crooks, A., and Radzikowski, J. (2011). Harvesting ambient geospatial information from social media feeds. *GeoJournal*, 78(2):319–338.

Stillerman, J. and Salcedo, R. (2012). Transposing the Urban to the Mall: Routes, Relationships, and Resistance in Two Santiago, Chile, Shopping Centers. *Journal of Contemporary Ethnography*, 41(3):309–336.

Sud, A., Andersen, E., Curtis, S., Lin, M., and Manocha, D. (2008). Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs. *IEEE Transactions on Visualization and Computer Graphics*, 14(3):526–538.

Tallevi-Diotallevi, S., Kotoulas, S., Foschini, L., Lcu, F., and Corradi, A. (2013). Real-Time Urban Monitoring in Dublin Using Semantic and Stream Technologies. In *The Semantic Web ISWC 2013*, pages 178–194. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-41338-4_12.

Tao, S., Manolopoulos, V., Rodriguez, S., and Rusu, A. (2012). Real-Time Urban Traffic State Estimation with A-GPS Mobile Phones as Probes. *Journal of Transportation Technologies*, 02(01):22.

Tim Berners-Lee, James Hendler, and Ora Lassila (2001). The Semantic Web. *Scientific American*, 284(5):34–43.

Tolani, D., Goswami, A., and Badler, N. I. (2000). Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. *Graphical Models*, 62(5):353–388.

Torrens, P. M. (2012). Moving Agent Pedestrians Through Space and Time. *Annals of the Association of American Geographers*, 102(1):35–66.

Torrens, P. M. (2014a). High-fidelity behaviours for model people on model streetscapes. *Annals of GIS*, 20(3):139–157.

Torrens, P. M. (2014b). High-resolution spacetime processes for agents at the builthuman interface of urban earthquakes. *International Journal of Geographical Information Science*, 28(5):964–986.

Torrens, P. M. (2015). Intertwining agents and environments. *Environmental Earth Sciences*, 74(10):7117–7131.

Torrens, P. M. (2016). Computational Streetscapes. *Computation*, 4(3):37.

Torrens, P. M., Nara, A., Li, X., Zhu, H., Griffin, W. A., and Brown, S. B. (2012). An extensible simulation environment and movement metrics for testing walking behavior in agent-based models. *Computers, Environment and Urban Systems*, 36(1):1–17.

Townsend, A. M. (2000). Life in the Real-Time City: Mobile Telephones and Urban Metabolism. *Journal of Urban Technology*, 7(2):85–104.

Townsend, A. M. (2013). *Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia*. W. W. Norton & Company.

Turner, A. (2009). The Role of Angularity in Route Choice. In Hornsby, K. S., Claramunt, C., Denis, M., and Ligozat, G., editors, *Spatial Information Theory: 9th International Conference, COSIT 2009 Aber Wrac'h, France, September 21-25, 2009 Proceedings*, pages 489–504. Springer Berlin Heidelberg, Berlin, Heidelberg.

Turner, A. and Penn, A. (2002). Encoding Natural Movement as an Agent-Based System: An Investigation into Human Pedestrian Behaviour in the Built Environment. *Environment and Planning B: Planning and Design*, 29(4):473–490.

Unity Technologies (2017). Unity - Manual: Navigation Areas and Costs.

Wagner, N. and Agrawal, V. (2014). An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster. *Expert Systems with Applications*, 41(6):2807–2815.

Whyte, W. H. (1980). *The Social Life of Small Urban Spaces*. Conservation Foundation.

Whyte, W. H. (1988). *City: Rediscovering the Center*. Anchor Books.

Wiener, J. M. and Mallot, H. A. (2003). 'Fine-to-Coarse' Route Planning and Navigation in Regionalized Environments. *Spatial Cognition & Computation*, 3(4):331–358.

Wiener, J. M., Schnee, A., and Mallot, H. A. (2004). Use and interaction of navigation strategies in regionalized environments. *Journal of Environmental Psychology*, 24(4):475–493.

Wilensky, U. (1997). NetLogo Segregation model. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Willis, A., Gjersoe, N., Havard, C., Kerridge, J., and Kukla, R. (2004). Human Movement Behaviour in Urban Spaces: Implications for the Design and Mod-

elling of Effective Pedestrian Environments. *Environment and Planning B: Planning and Design*, 31(6):805–828.

Wolfram, S. (1984). Computation theory of cellular automata. *Communications in Mathematical Physics*, 96(1):15–57.

Wooldridge, M. and Jennings, N. R. (1998). Pitfalls of Agent-oriented Development. In *Proceedings of the Second International Conference on Autonomous Agents*, AGENTS '98, pages 385–391, New York, NY, USA. ACM.

Woolley, H. and Johns, R. (2001). Skateboarding: The City as a Playground. *Journal of Urban Design*, 6(2):211–230.

Wu, B. M. and Birkin, M. H. (2012). Agent-Based Extensions to a Spatial Microsimulation Model of Demographic Change. In Heppenstall, A. J., Crooks, A. T., See, L. M., and Batty, M., editors, *Agent-Based Models of Geographical Systems*, pages 347–360. Springer Netherlands. DOI: 10.1007/978-90-481-8927-4_16.

Yu, Q. and Terzopoulos, D. (2007). A decision network framework for the behavioral animation of virtual humans. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '07, pages 119–128, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

Zheng, X., Zhong, T., and Liu, M. (2009). Modeling crowd evacuation of a building based on seven methodological approaches. *Building and Environment*, 44(3):437–445.

Zhou, B., Wang, X., and Tang, X. (2012). Understanding collective crowd behaviors: Learning a Mixture model of Dynamic pedestrian-Agents. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2871–2878.

Zuccato, E., Chiabrando, C., Castiglioni, S., Bagnati, R., and Fanelli, R. (2008). Estimating Community Drug Abuse by Wastewater Analysis. *Environmental Health Perspectives*, 116(8):1027–32.