SCHOOL OF GEOGRAPHY
FACULTY OF ENVIRONMENT

UNIVERSITY OF LEEDS

Evaluating an Agent-Based Model of Burglary

Working Paper 10/1

Nick Malleson

Andrew Evans

Alison Heppenstall

Linda See

School of Geography, University of Leeds, LS2 9JT

February 2010

Contents

Abstract

An essential part of any modelling research is to evaluate how a model performs. This paper will outline the process of evaluating a new agent-based model that is being developed to predict rates of residential burglary. The model contains a highly detailed environment which is representative of Leeds, UK. Following Castle and Crooks (2006), the process of evaluating the model will be segregated into three distinct activities: verification, calibration and validation. Verification refers to the process of establishing whether or not the model has been built correctly. This can be an extremely difficult process with complex models. Here, verification is accomplished by "plugging-in" different types of virtual environment which enables the researcher to limit environmental complexity and thus isolate the part of the model that is being tested. Following verification, calibration is the process of configuring the model parameters so that the output match some field conditions. However, this is a non-trivial task with models that are inherently spatial as it must be decided *how* to compare the two data sets. To this end, the paper will explore a number of spatial techniques and statistics that can be used to compare spatial data before documenting the process of calibrating the model. After calibration, it is necessary to ensure that the model has not been *over-fitted* to the calibration data (a process termed *validation*). Here, this is accomplished by running the model using environmental data from a different time period and comparing the results to the corresponding crime data.

1 Introduction

An essential part of any modelling research is to evaluate how a model performs. This paper will outline the process of validating a new agent-based model that is being developed to predict rates of residential burglary. The means of evaluation include the adaption of a method to compare point datasets and the use of varying types of environment so that model complexity can be increased incrementally.

The model itself consists of numerous burglar agents who are controlled by a cognitive framework and are able to socialise, undertake legitimate employment and to consume drugs. Each agent also has a home which they sleep in. The agents are able to navigate a detailed virtual environment which consists of roads, communities and houses, and is representative of a real city (Leeds, UK, in this case). As the model runs, the agents are able to make decisions about what action they will take and, if the need arrises, can perform burglary to generate wealth. For more information about the model, refer to Malleson et al. (2009a), Malleson et al. (2009b), Malleson and Brantingham (2009) or Malleson (2008). The research is a PhD project at the School of Geography, University of Leeds entitled "Agent-Based Modelling of Burglary".

Following Castle and Crooks (2006), the process of evaluating the model will be segregated into three distinct activities: *verification*, *calibration* and *validation*. This section will discuss these processes and outline how they will be used. Section 2 will then discuss how sets of spatial data can be compared as this is relevant throughout the remainder of the paper. This is followed by Section 3 which will outline the process of verifying the model and testing the sensitivity of the model to some parameters. Section 4 will discuss how the model was calibrated and finally Section 5 will validate the model against a separate data set to that which was used for calibration.

1.1 Verification

Verification refers to the process of ensuring that the model has been programmed correctly and behaves as it is expected to. This can be thought of as "inner validity" (Axelrod, 1997; Brown, 2005). The model produced for this research is highly complicated; it contains numerous environmental processes, detailed human behaviour, a large number of agents and intricate interactions between the agents and the environment. Also, from a programming perspective, it contains more than 10,000 lines of computer code and over 100 classes. For all of these reasons verification is non-trivial. A reliable method of confirming that a simulation has been programmed correctly is to re-implement the same model separately, ideally using a different programming language. This process can be termed "docking" (Axtell et al., 1996). However, this is far from feasible for this project because re-implementing such a large model would be extremely time consuming. Instead, the model has been developed with the capacity for different types of environment to be "plugged-in" without having to change the implementation of other parts of the model (such as the agents, houses, communities etc.). The three types of environment that have been implemented are:

- A highly detailed GIS (vector) environment that closely represents real buildings, roads and neighbourhood in the study area;
- A cellular grid (raster) environment which also contains buildings, roads and communities but can be specified exactly by the reserach and has considerably less environmental complexity;
- A non-spatial environment in which every journey time can be specified exactly, regardless of the places that the agent is travelling from and to. This is the least complex type of environment and enables the

researcher to separate the model from any spatial complexity.

The main aim of this novel feature is to allow environmental complexity to be added incrementally, making it easier to isolate any errors. Although this approach is not as comprehensive as complete docking, it is an effective substitute. Equally importantly, the approach addresses a worry in the environmental criminology field that models with complex environments can actually detract from the purposes of modelling; which is to better understand the dynamics of the system (Elffers and van Baal, 2008). By increasing environmental complexity gradually the researcher is able to both fully understand the dynamics of the model before incrementing the complexity of the environment and to isolate parts of the model from the effects created by geographical space.

1.2 Calibration

Calibrating a model is the process of configuring the model parameters with the aim of producing output that matches some field conditions. Often this is accomplished by comparing model results to an expected data set (recorded burglaries in this case) and configuring the parameters to provide the lowest error between the simulated and expected data. Automatic methods which vary numerical parameters systematically in search of the optimal configuration are used regularly (Heppenstall et al., 2006; Malleson et al., 2009). However, the complexity of this model does not allow for such neat calibration techniques for three reasons: firstly, several parameters cannot be translated to a ranged numerical scale (e.g. the starting locations of the agents); secondly because (as Section 2 will illustrate) determining the goodness-of-fit between simulated and expected data is non-trivial for spatial models; and thirdly, the considerable computation time of the model deems some methods impossible to run on available hardware (even when utilising a high-performance computing grid). Therefore in the absence of formal calibration methods, model calibration will be performed manually through analysis of results and varying parameters based on the theory and knowledge of the dynamics of the model.

1.3 Validation

Model validity is the extent to which the model is able to represent the system it is attempting to simulate (Casti, 1997). Commonly, validation refers to the process of applying the model to a new system other than the one it was calibrated on. If the model is still able to perform reasonably then it indicates that the model has not been over-fitted to the original data set and can be safely applied to different scenarios. Here, validation poses somewhat of a problem because although there is sufficient crime data available, this is not the case for environmental data. Large parts of the model are based on the UK census which is only available for 2001. Section 5 will discuss how this problem can be mediated.

2 Comparing Spatial Data

A large part of any spatial modelling research will be spent comparing simulation results to each other and to expected data. This section will review and test different methods of both describing and then comparing point patterns with the aim of determining which is the best for comparing model data to field data. To test the different point comparison methods available throughout this section, three point patterns will be used which are all based in an area that covers part of the south-east of Leeds, UK:

- **Model1**: Sample model data. An example of simulated data.
- **Model2**: Sample model data. An identical simulation to the one used to generate the Model1 data set. Although there are small differences in the data (i.e. the model is stochastic) the data are generally similar (as illustrated by Figure 1).
- **Random**: A completely random data set, generated by a simulation of a random spatial process. Point patterns such as these are said to exhibit complete spatial randomness (O'Sullivan and Unwin, 2003), i.e. they are produced by a process which is completely random spatially.

The three data sets are illustrated by Figure 1 using the kernel-density algorithm to generate the raster surfaces (the algorithm is discussed in more detail in Section 2.1.2). The ArcGIS "Create Random Points" tool (ESRI, 2006) was used to generate the random points within the simulation boundary. Each data set contains the same number of points. For most of the methods used in this section, changing the number of points does not influence the results so long as the distribution of the data is the same. Where different numbers of points *will* affect results this will be discussed explicitly.

2.1 Comparing Point Patterns Visually

Simply displaying point data on a map is not generally informative of the overall pattern, particularly for patterns which contain large numbers of points. For example, in Figure 1 it is extremely difficult to discern how dense the point patterns are by purely looking at the locations of the individual points. It is unclear where points overlap and identifying the density relies on human objectivity. Another problem, which is unrelated but no less significant, is that data protection or privacy rules often forbid point locations from being displayed because they will identify individuals. In crime studies such as this one, this is often a problem because points regularly centre on individual houses. Therefore there are alternative methods which can be used to describe point patterns that are both more informative and will not identify individual locations.

2.1.1 Aggregating up to Administrative Areas

An alternative to displaying or analysing "raw" point data is to first aggregate the points up to a larger area. Then it is possible to analyse the aggregated data and produce thematic maps of the counts of points in each area or calculate rates. This is the approach taken in similar research (Kongmuang, 2006; Shepherd, 2006) and also by numerous police services who produce on-line crime maps for the public, e.g. Metropolitan Police (2009). However, as illustrated from the test data in Figure 2, this approach clearly suffers from the ecological fallacy which is evident by the degree that the maps differ depending on the scale of the boundaries used. It is also likely that the method will suffer from the modifiable areal unit problem as it exists in all studies which make

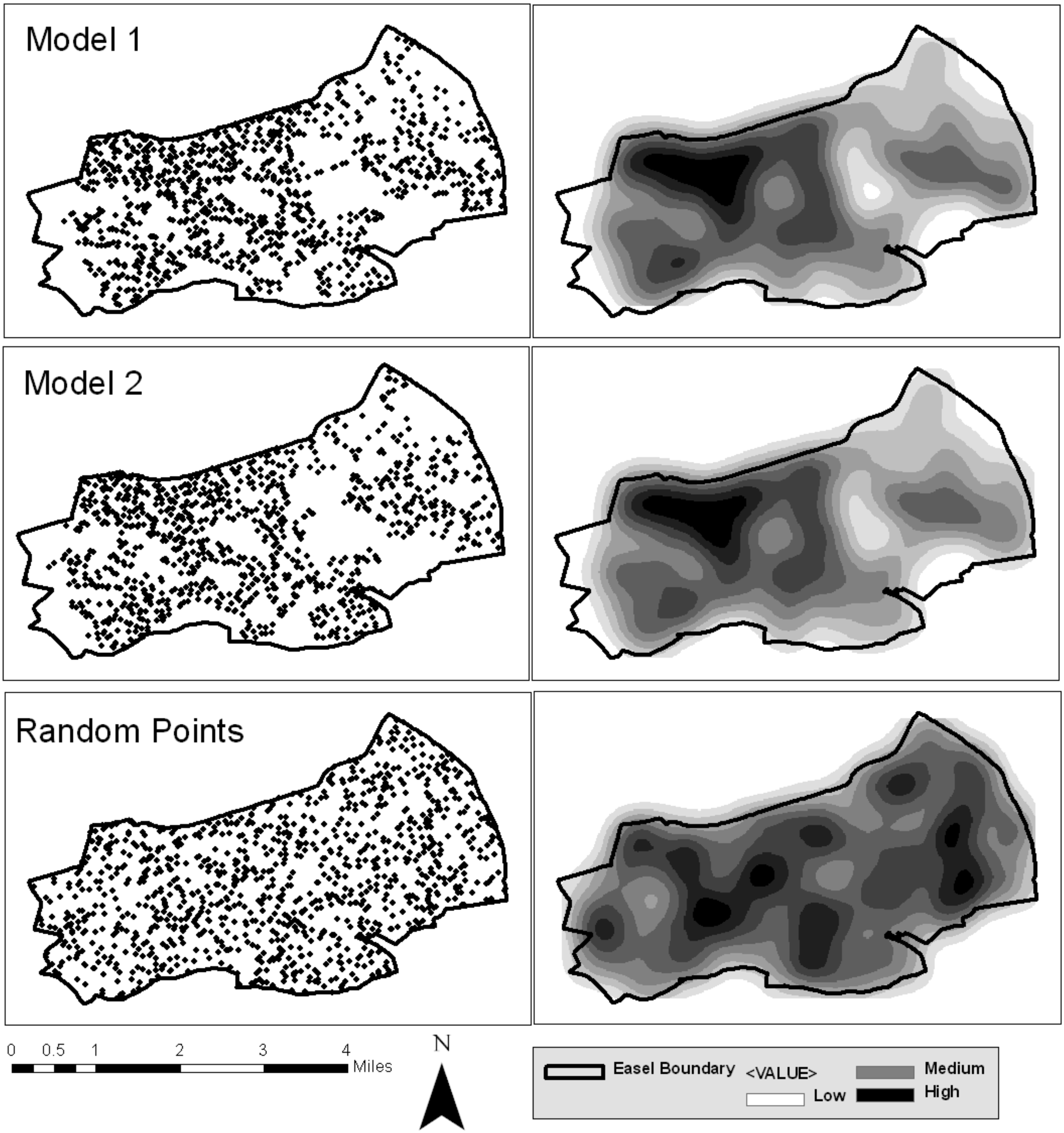

Figure 1: The three point data sets which will be used to experiment with spatial methods throughout the remainder of this section.

use of aggregated data (Wrigley, 1995). Furthermore, choosing too large a boundary (such as the medium Super Output Area) hides important intra-area differences, seriously mis-representing the underlying pattern (note the similarity between model and random data at the largest area boundary). Experimenting with using rates rather than absolute counts were no more promising.

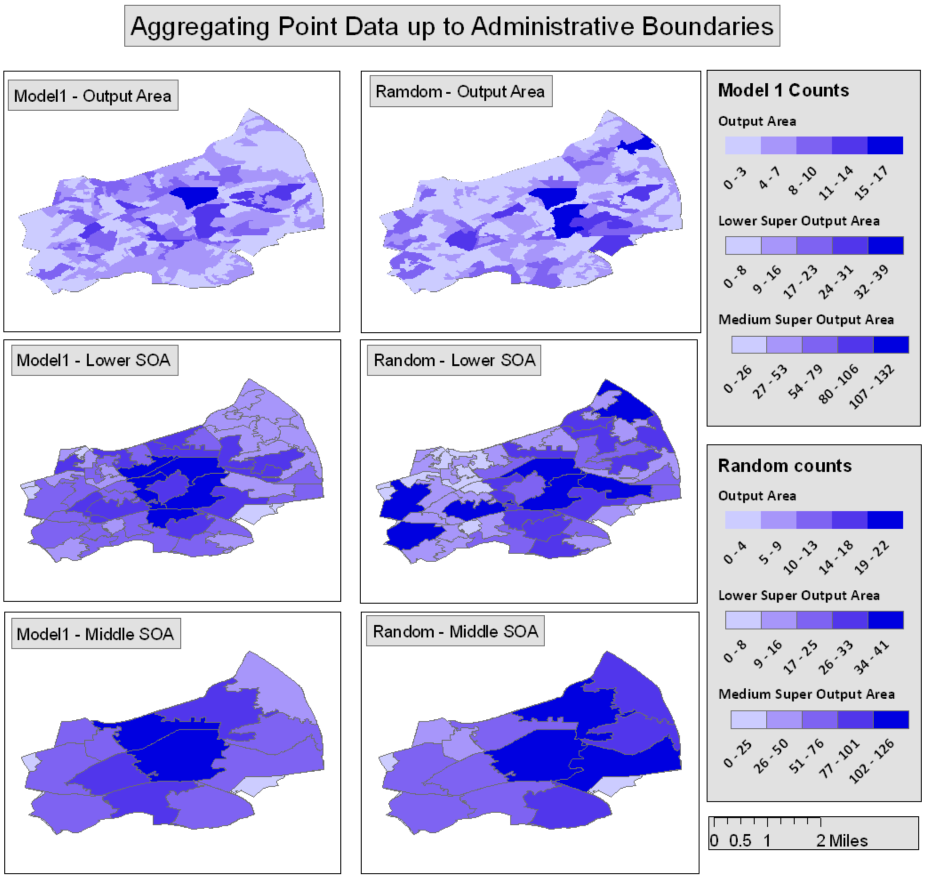

Figure 2: Counts of points aggregated up to (Super) Output Area administrative boundaries.

For the reasons given above, these types of maps are generally considered extremely misleading by crime analysts and should not be used to illustrate crime patterns (Chainey, 2009a). A solution to these problems can be sought by aggregating up to arbitrary areas rather than those defined for specific administrative purposes. Figure 3 illustrates counts of the data aggregated to various regular grids. The first 200m grid is shifted 100m south west to test for the modifiable areal unit problem and then increased to a 400m resolution to test for the ecological fallacy.

The first thing to notice is how different the grid-aggregated maps are compared to the OA-aggregated maps. Crime hotspots appear to be in very different places. By examining the first two grids for the Model1 data set it is apparent that the method *does* suffer from the modifiable areal unit problem, although not significantly as the general pattern is similar. This finding is consistent with the crime mapping literature (Chainey and Ratcliffe, 2005). Also, the ecological fallacy is somewhat reduced as the general pattern using a 400m resolution is similar to the two 200m resolution grids. Eck et al. (2005) found that, assuming an appropriate resolution had been set, grid thematic mapping was better at representing crime hotspots than administrative area aggregation.

It must be noted that the results for the random data are much less promising as all three grids look very different. This is to be expected though because, as was illustrated earlier, the random data are less clustered.

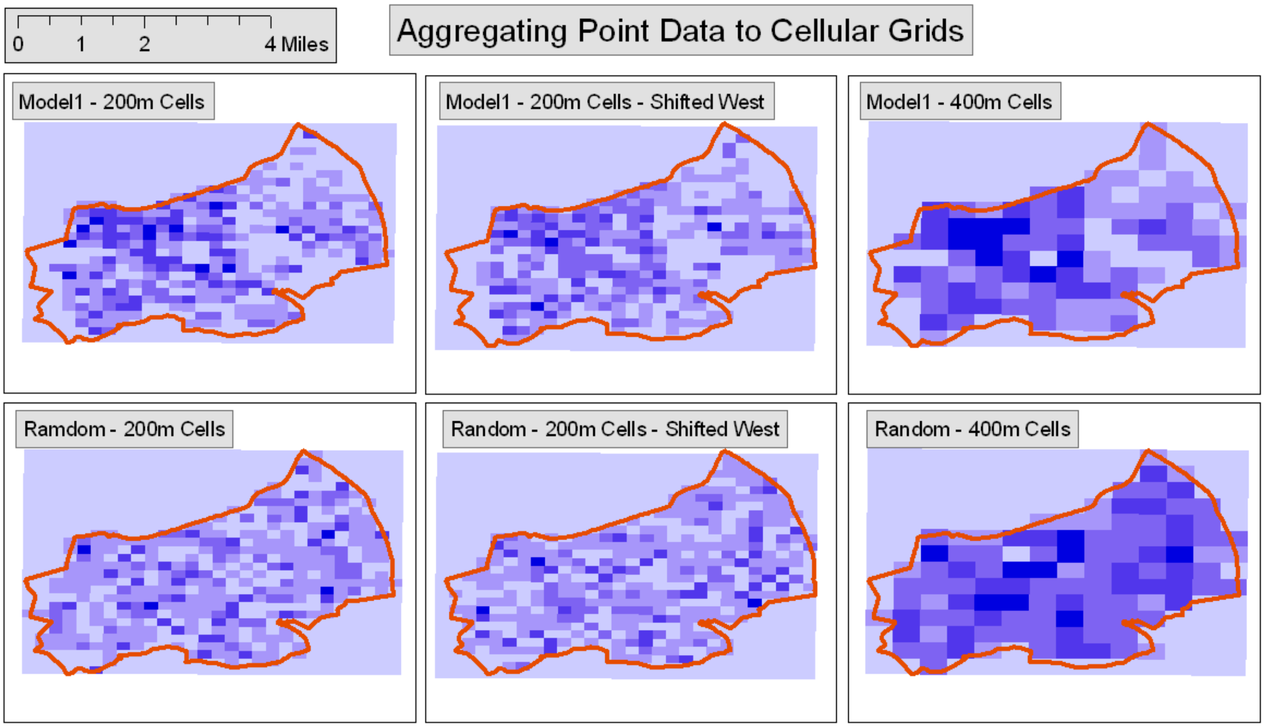

Figure 3: Point patterns aggregated to various regular grids. The first 200m grid is shifted 100m south west to test for the modifiable areal unit problem.

As crime data (as with most natural processes) usually show some degree of clustering the poor representation of the random data set is not seen to be a problem.

A note about cartographic thresholds

It should be noted that the choice of thematic thresholds is extremely important as these can significantly effect the conclusions drawn from a map. For example, Figure 4 illustrates the same data (Mode1 counts aggregated to Lower SOA) using four commonly-used thematic classification methods:

- **Natural Breaks (Jenks)**. Sub-ranges are based on the "natural groupings inherent in the data" (ESRI, 2006) by attempting to group similar values and maximising the difference between classes.
- **Equal Interval**. Each interval is given the same length.
- **Equal Count**. Each interval holds the same number of observations (or as near as possible).
- **Geometric Interval**.Sub-ranges are based on class intervals that have a geometrical series (ESRI, 2006).

Of the four illustrated, all appart from equal interval seem to group the larger counts into a single sub-range. This can be very misleading and fails to pinpoint the areas that have extreme values which are often the most interesting. Furthermore, the equal interval method is the easiest to interpret and will be the most consistent when generating maps from different data. Therefore this is the method which will be used throughout the research unless otherwise stated.

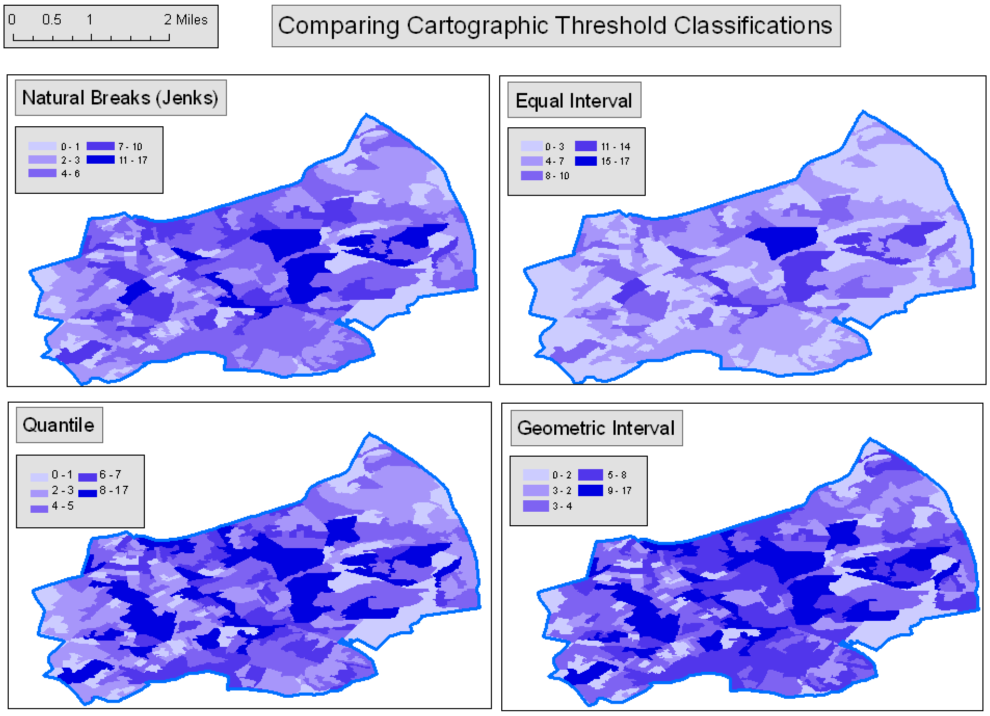

Figure 4: Comparing different thematic classification methods.

2.1.2 Creating density surfaces

The previous section illustrated that aggregating points to administrative boundaries is not a reliable method for crime clusters. Grid thematic mapping was introduced as an alternative, but the "blockiness" of grid thematic maps makes them less visually appealing which can be a problem if crime maps need to be shown to stakeholders (Chainey and Ratcliffe, 2005).

An alternative approach, which is becoming the most commonly used by crime analysts and is seen as the most reliable (Chainey, 2009b), is the use of density estimation algorithms. Kernel density estimation (KDE) methods are commonly used to estimate the density of points at every location in a study region. They operate by counting the number of points within a given distance (or 'kernel'). O'Sullivan and Unwin (2003) describe a "naive" method to estimate the density at a point, p, as:

$$density_p = \frac{\#(S \in C(p,d))}{\pi d^2} \tag{1}$$

where d is the kernel distance, $C(p,d)$ is a circle of radius d centred at point p, S is the set of all points and # means the "number of" (as in Bailey and Gatrell (1995)). It is common to place a square grid over the study region and then estimate the density at the centre of every cell. Figure 5 illustrates that the size of the kernel (d) is extremely important but the cell size only changes the resolution of the map, not the actual density. Values of d that are too large can result in 'over-smoothing' where two separate clusters appear as one. Similarly, values that are too small result in a single cluster around each point, also failing to capture the locations of clusters or the overall point density. In extreme cases, large d values result in similar density estimates for all cells (which will be close to the average point density) and small values result in zero density for all cells but those that have a point in them (O'Sullivan and Unwin, 2003). It is therefore advisable to experiment with bandwidths and to choose appropriately for the phenomenon under investigation. There are various methods which can be used to improve density estimations and the most common is to weight closer events more highly than distant events. In most analyses performed here, a quadratic kernel function is used as described in Silverman (1986) and implemented in the ArcGIS software package (ESRI, 2006).

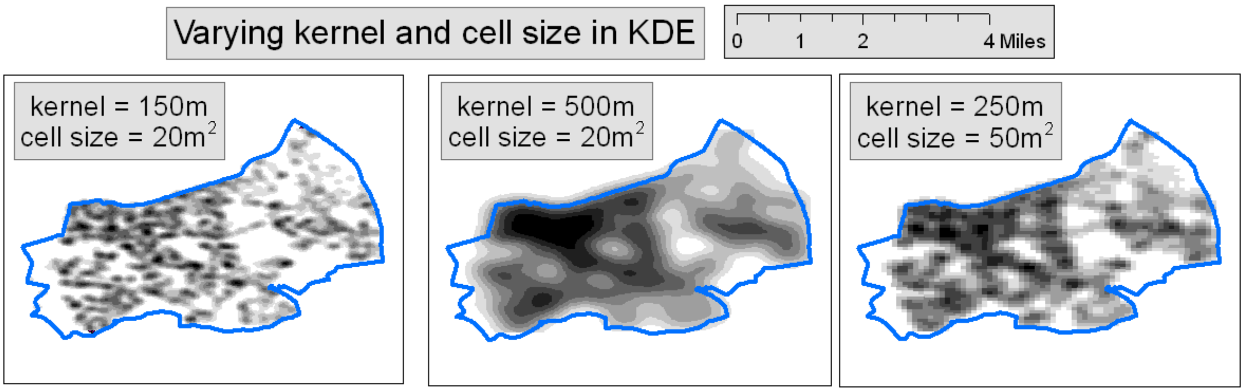

Figure 5: Examples of the effects of changing the size of the kernel and the grid cell size used in the KDE algorithm on the Model1 point pattern.

There are other methods which can be used to pinpoint hotspots visually. In particular, local indicators of spatial association (LISA) methods examine each area in a map to determine whether or not the area's value is significantly greater (or less) than that of its neighbours. Arguably the most promising LISA static, but still very under-used in the field of crime mapping (Chainey, 2009a) is the GI* statistic (Getis and Ord, 1992, 1996). GI* (and the similar GI statistic[1]) has the advantage that it is able to provide a significance test which indicates how *statistically significant* a potential hotspot is. Using this information, thematic thresholds can be generated from the significance levels which allows the thematic thresholds to remain consistent throughout all maps. Section 2.1.1 noted that thematic thresholds are an extremely important element to any map and can drastically change the appearance of data. A detailed investigation into the GI* static is unnecessary because (as will be shown) it will not ultimately be used, but for the interested reader more information can be found in Getis and Ord (1992).

Figure 6 illustrates the results of the GI* statistic applied to the Model1 point pattern, aggregated to 100m and 200m (cell length) grids. The 100m grid does not appear to capture the spatial structure of the data well, there are numerous empty cells surrounding a few with a single burglary in them which makes them appear significant. The 200m resolution grid is more promising and, as far as pinpointing hotspots, has many similarities with the KDE surfaces. It is likely that this map is, in fact, better at pinpointing hotspots than the KDE surface. However, the results are less visually pleasing and it is more difficult to interpret the overall structure of the underlying point pattern using the GI* map than it is with the KDE surfaces. Therefore, as density maps will not be used to perform any numerical analysis (they will only be used for visual data comparisons) the fact that the GI* maps can more accurately pinpoint hotspots is not a good enough reason to use them over KDE surfaces.

[1]The difference between GI and GI* is that GI does not include the area being examined in its neighbourhood calculation whereas GI* does

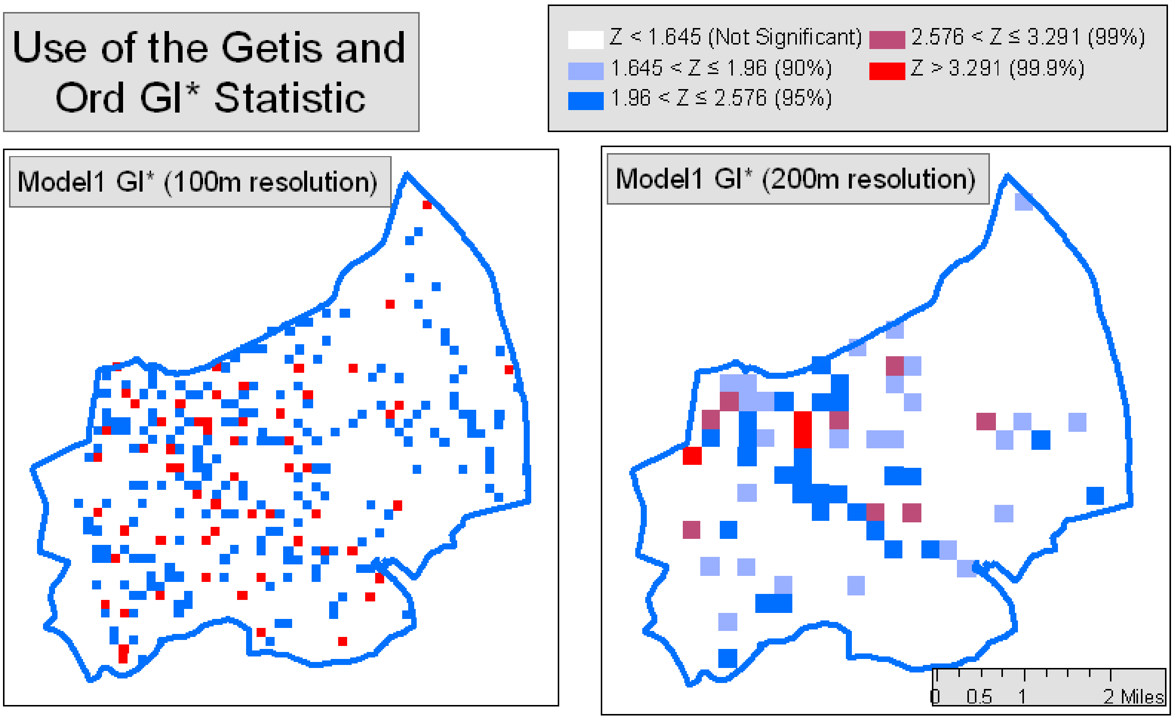

Figure 6: The GI* (Getis and Ord, 1992) static applied to grid-aggregated Model1 points.

2.2 Comparing Point Patterns Mathematically

Whilst methods to compare point patterns visually are essential, there is still much which remains susceptible to human objectivity. The maps used previously are able to indicate which patterns *look* similar, but this on its own does not provide sufficient evidence for (dis)similarity. This section will outline methods that can be used to describe the structure of points patterns mathematically and compare them to others.

2.2.1 The Nearest Neighbour Index / R Statistic

Nearest neighbour methods of analysis involve exploring patterns by comparing distances from a given point to its nearest neighbour, whether these distances are taken from within the same data set or between two different ones (such as comparing an observed point pattern with one produced by a random process (Fotheringham and Rogerson, 1994)) .

The mean minimum nearest neighbour distance, $\bar{d}_{min}$, in a point pattern is calculated by averaging the distances between each point, i, and its nearest neighbour, j:

$$\bar{d}_{min} = \sum_{i=1}^{n} \frac{d_{ij}}{n} \tag{2}$$

where n is the number of points. It is then possible to compare $\bar{d}_{min}$ to an expected mean minimum distance which would be found in a randomly generated data set, $\bar{\delta}$, defined by Clark and Evans (1954) as:

$$\bar{\delta} = \frac{1}{2\sqrt{A/n}} \tag{3}$$

where A is the total area under study (so A/n is the average point density). The nearest neighbour index (NNI), which was originally formulated as the Clark and Evans R statistic (Clark and Evans, 1954), can then be defined as the ratio of the observed mean minimum distance to the expected one:

$$R = \frac{\bar{d}_{min}}{\bar{\delta}} \tag{4}$$

The range of the statistic is from 0, which indicates that all points are at the same location, to 2.15, suggesting uniformity (Chainey and Ratcliffe, 2005). Values of $R < 1$ suggest that the mean minimum distance is lower than expected under CSR and the data are therefore clustered. To improve the strength of the analysis, Bailey and Gatrell (1995) offer a significance test that has also been implemented in CrimeStat (Levine, 2006).

The major problem with the R statistic is accounting for edge effects and although there are some solutions (such as the circular or rectangular correction algorithms built into the CrimeStat application (Levine, 2006)) these are not ideal (Chainey and Ratcliffe, 2005). Nevertheless the statistic can still be used to give a general picture of whether or not clustering is present and is therefore useful as a preliminary procedure (Bailey and Gatrell, 1995)

Table 1 illustrates the NNI values for the three data sets along with their Z scores and significance levels. The Z values for Model1 and Model2 data suggest that both are significantly clustered at the 99.99% level. Importantly, this implies that the model data are clustered so there is some utility in performing hotspot analysis later. The random data appears to be slightly more uniform than would be expected, but the R value is close enough to 1.0 not to be of concern. The result is probably somewhat due to edge effects, as using border correction routines built into CrimeStat reduces the significance to from 99% to 95%.

Table 1: Nearest Neighbour Index, also known as Clark and Evans R (Clark and Evans, 1954), values for the three test data sets. The square area used was 6.66905 mi^2 and rectangular boundary correction was applied using the CrimeStat application.

Data Set	d_{obs} (m)	d_{exp} (m)	NNI (R)	Z	p-value (two tailed)
Model 1	54.9063	64.3332	0.85347	-9.1838	0.0001
Model 2	53.0307	64.3332	0.82431	10.8649	0.0001
Random	67.6524	64.3332	1.05160	3.1908	0.01

2.2.2 Nearest Neighbour Hierarchical Clustering

Hierarchical cluster-hunting algorithms are a common method of searching for clusters in data and these can be utilised spatially using the idea of nearest neighbours. Hierarchical methods work in stages, determining at each iteration whether or not data should be grouped into a cluster. Figure 7 provides an example of an agglomerative, or "bottom-up" clustering routine. Each data point is compared to every other point and if a number of points are within a given distance of each other they are grouped into 1st order clusters. The algorithm then repeats this step for the 1st order clusters, grouping them appropriately to second order clusters, and continues until either no more clusters can be generated or until every point is a member of a single cluster.

It is possible to vary the minimum distance between points which form a group and the minimum number of points which must be grouped to form a cluster. Figure 8 illustrates the results of two an NNH algorithms applied using the CrimeStat (Levine, 2006) application and altering the minimum number of points per cluster

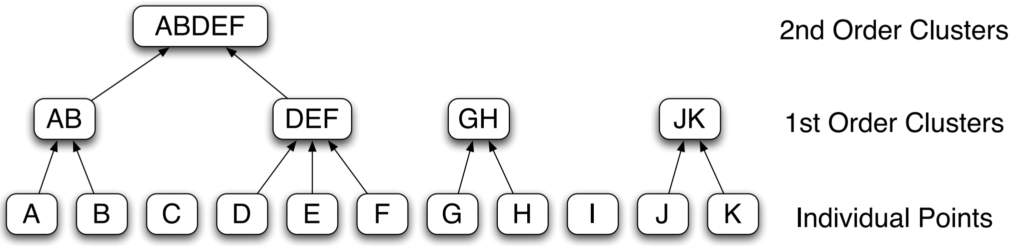

Figure 7: An example of a agglomerative hierarchical clustering routine. Divisive routines also exist which are the opposite to agglomerative ones whereby every data point starts in one large cluster and points are removed iteratively

from 10 to 5. The minimum distance between points for them to form a group is variable and based on the probability that the points would be grouped if the pattern were produced under CSR, for more details see Levine (2009). Figure 8 demonstrates that the results of the algorithm can be very different depending on the values of the parameters chosen. This problem coupled with the fact that it is difficult to establish reliable, non-arbitrary values for parameters means that this algorithm will be rarely used.

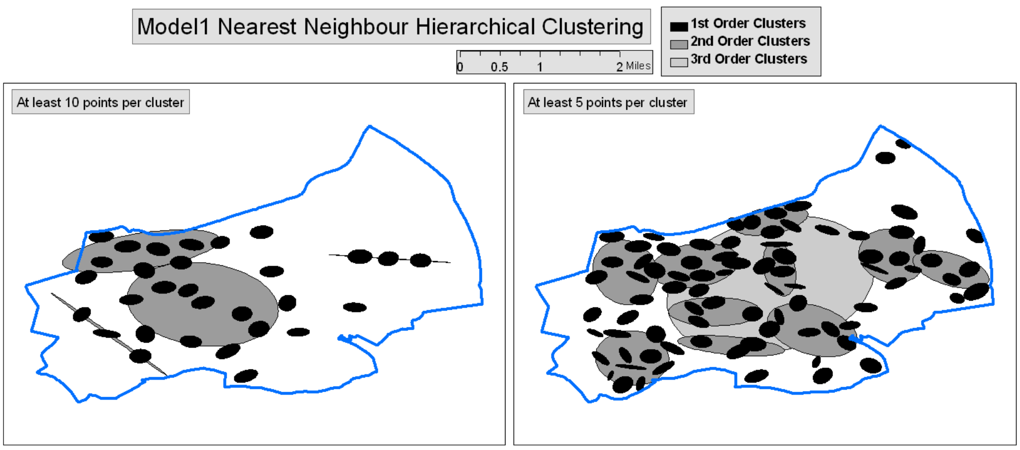

Figure 8: Clusters generated by the Nearest Neighbour Hierarchical (NNH) algorithm on model1 data by varying the minimum number of points required to form a cluster.

2.2.3 The G and F functions

A drawback with mean nearest-neighbour functions (e.g. NNI) is that although summarising an entire point pattern to a single mean is concise, it is too simplistic to be really useful. An alternative to these are the G and F functions which estimate the cumulative frequency distribution of the nearest-neighbour distances (O'Sullivan and Unwin, 2003). The G function, at a given distance, d, is the fraction of points, s_i, whose nearest neighbour is less than d away:

$$G(d) = \frac{\#(d_{min}(s_i) < d)}{n} \tag{5}$$

where # means "the number of" (as in Bailey and Gatrell (1995)) and n is the total number of points. The

F function is similar to G, but as an alternative to using the nearest neighbour distance from each point, it uses the distance from a randomly selected location anywhere on the map. As defined by O'Sullivan and Unwin (2003): if $\{p_1...p_i...p_m\}$ is a set of m randomly selected locations and S is the set of all points, then

$$F(d) = \frac{\#(d_{min}(p_i, S) < d)}{m} \tag{6}$$

Graphs of F and G against d describe the clustering of the data at different distances and can be used to differentiate between clustered and uniform data (see O'Sullivan and Unwin (2003) for a fuller discussion)). In all graphs illustrated here the input distances are a list of sequential numbers ranging from 0 to the maximum possible distance in the study area.

Figure 9 illustrates the G functions for the data sets. The Model1 and Model2 functions rise more sharply than that of the Random points which implies that more points have close nearest neighbours in the model data than the Random data. This is indicative of clustering. This graph therefore further supports the results of the NNI test, suggesting that the model data are more clustered than random and that Model1 and Model2 data sets are similar.

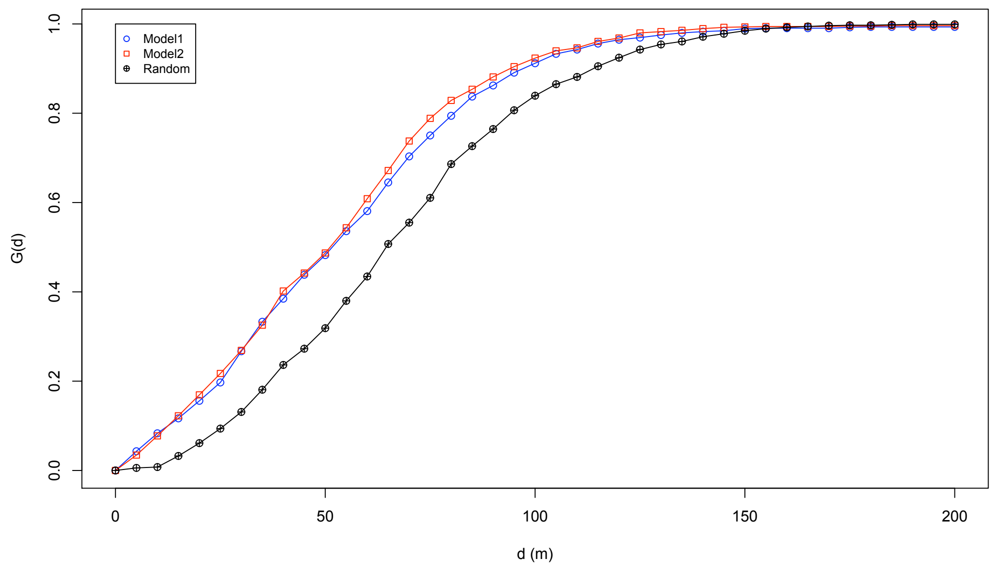

Figure 9: A graph of G(d) for the test points data

2.2.4 Ripley's K Function

A failing of the NNI and the G and F functions is that they only consider the single nearest neighbour distance in their calculations so disregard a considerable amount of information. Ripley's K function (Ripley, 1977), on the other hand, takes all the neighbours that are within a given distance into account. For each point, s_i, the number of neighbours within a given distance, d, is counted. The value of K at distance d is then calculated as the mean of all counts divided by the overall point density, defined by O'Sullivan and Unwin (2003) as:

$$K(d) = \frac{\sum_{i=1}^{n} \#(S \in C(s_i, d))}{A/n} \tag{7}$$

where $C(s_i, d)$ is a circle of radius, d, centred at point s_i and A is the size of the area being studied.

As with the G and F functions, graphs of $K(d)$ reveal information about the clustering of the points. For example, Figure 10, from O'Sullivan and Unwin (2003), illustrates the graphs of $K(d)$ for hypothetical clustered and un-clustered data. Observing the clustered data, the $K(d)$ value at approximately $d = 0.2$ indicates the diameter of the clusters and values above $d = 0.6$ is indicative of the cluster separation.

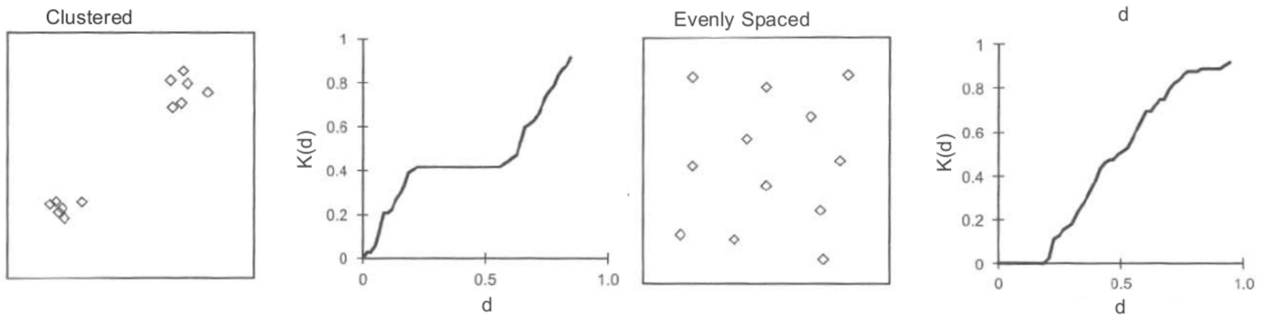

Figure 10: Graphs of $K(d)$ for clustered and un-clustered data. Taken from O'Sullivan and Unwin (2003, page 94).

Figure 11 illustrates graphs for the test data sets. It becomes apparent that graphs generated from real data with large numbers of points are much more difficult to interpret than the simple example provided by O'Sullivan and Unwin (2003). The form of the curve is similar to the evenly spaced example which suggests that the data are not clustered, although previous analyses have provided strong evidence to suggest otherwise. It appears, therefore, that the K function in isolation does not provide any useful information about a data set for this research. However, there is *some* utility in comparing K functions for different data.

It is surprising that the K functions appear so similar as the analyses thus far have all suggested that the model and random data sets are different. However, Figure 11 is somewhat misleading. Figure 12 illustrates graphs the differences in K functions by deducting one from another. It becomes apparent that the model data sets are much more similar to each other than they are to the random points, even if the larger maximum distance in the model is ignored (i.e. where $d < 4km$ there are still differences between the random and model data). This implies that when comparing different datasets it is more appropriate to graph the *difference* in their K functions rather than the K functions themselves.

2.2.5 The L Function for Ripley's K

A drawback of the K function is that it is informal, it *suggests* how the data are clustered but does not provide any formal evidence for whether or not this clustering is more or less than would be expected under CSR. However, it is possible to determine how the G, F and K functions behave under CSR and then use this information to determine whether or not the observed values are different. O'Sullivan and Unwin (2003) describe how the K function behaves under CSR and state that the L function is an appropriate transformation to indicate whether or not a given $K(d)$ value is greater or less than that which would be expected:

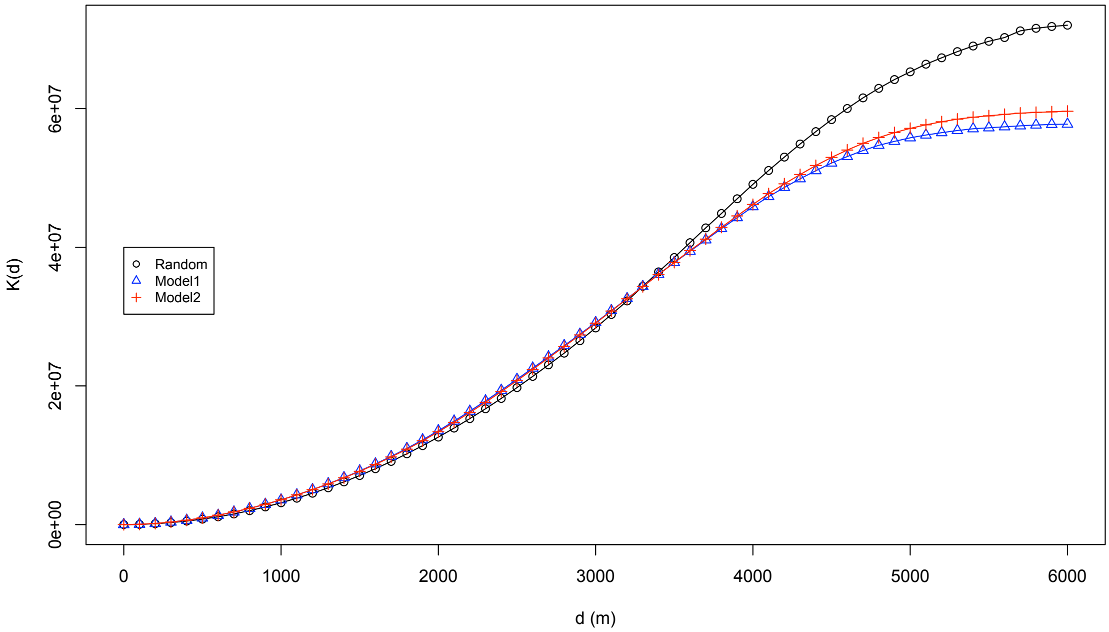

Figure 11: K(d) functions for model and random data sets.

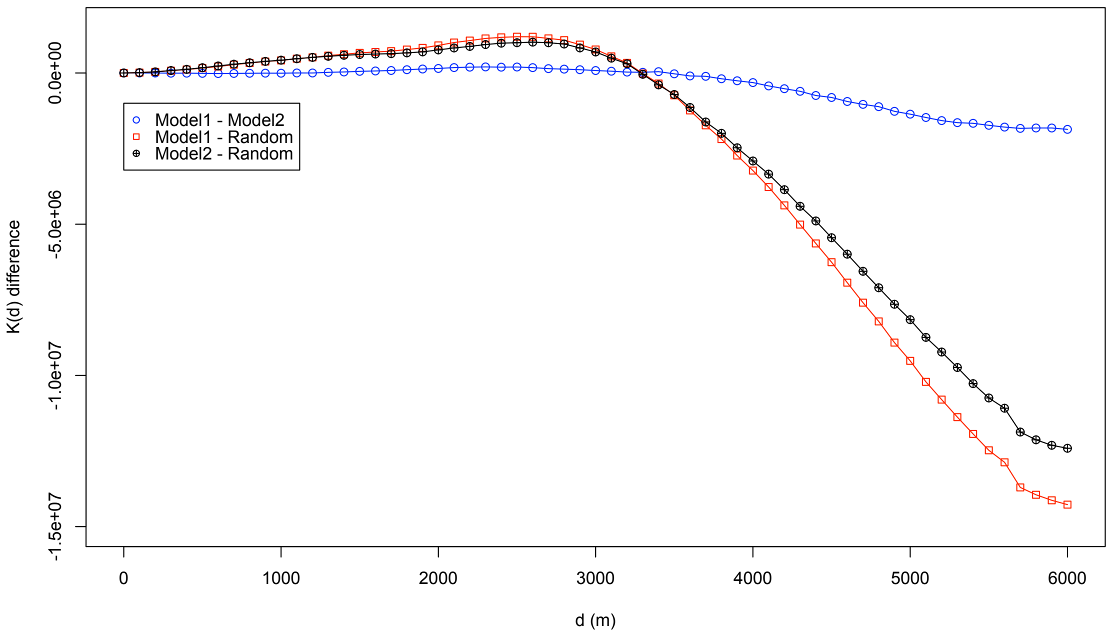

Figure 12: The difference in K(d) functions for model and random data sets.

$$L(d) = \sqrt{\frac{K(d)}{\pi} - d} \tag{8}$$

Values of $L(d) < 0$ suggest that there are fewer events in the space than would be expected under CSR and that the data are therefore less clustered. The reverse is true for $L(d) > 0$. It is important to note that using CSR as a baseline often provides little further insight because almost all natural processes do *not* exhibit spatial randomness. A better approach, which is commonly used in the field of crime mapping, is to compare graphs of $L(d)$ to determine how similar their clustering is. For example, comparing graphs of burglary and known drug dealer locations provides evidence for or against the hypothesis that burglary clusters around known drug dealers. Similarly, a comparison of $L(d)$ graphs for knife crime and stop-and-search occurrences can provide the police with evidence for how well their crime reduction initiatives are being targeted at crime hostpots (Chainey, 2009a).

Following this methodology, Figure 13 illustrates the $L(d)$ values for the test data sets. Values of $L(d) > 0$ suggest that the model data are more clustered than one would expect under CSR. The random point pattern equates to approximately $L(d) = 0$ for low d values which is to be expected as it is a simulation of CSR. Above $d \approx 3000$ the value of L begins to fall due to boundary effects: many of the large circles produced by the underlying K function are nearly empty at large distances because they cover areas outside the simulation boundary with no points (O'Sullivan and Unwin, 2003). The two L functions for Model1 and Model2 are very similar which provides more evidence that the point patterns are similar. They both have values of $L(d) > 0$ suggesting clustering. However, the techniques are shown to be extremely sensitive to boundary conditions and must, therefore, be used with caution. With the proviso that graphs taken in isolation are unreliable, the most useful aspect of these analyses for the project is their use in *comparing* data sets (i.e. whether or not the data sets are similar), not for making statements about their structure as a whole.

2.2.6 Raster Map comparisons

Other than comparing mathematical descriptions of clustering in point patterns, it is also possible to compare visual descriptions of point patterns mathematically. The KDE algorithm was outlined in Section 2.1.2 and can be used to generate a density surface from a point pattern. Once a raster surface has been generated, it can be classified to create a categorical raster map. Then it is possible to use one of numerous methods which can be used to compare categorical raster maps, particularly in fields such as landscape modelling. For a review of recent approaches, the reader is directed to Kuhnert et al. (2005). A brief example of the type of analysis that is possible will follow. It will be shown that these methods in general are not ideal for this project so the example method chosen is not detailed in depth.

Hagen (2003) has outlined a "fuzzy" enhancement to the Kappa statistic, known as Fuzzy Kappa, K_{fuzzy}, where fuzziness refers to the level of uncertainty in a map. Hagen notes that uncertainty can arise in two forms: spatial uncertainty (where a particular location in map A might be in a similar, but not identical, position in map B) and categorical uncertainty (where some categories might be more similar to each other than to other categories). Using fuzzy set theory, Hagen is able to produce local estimates of similarity and a global value:

$$K_{fuzzy} = \frac{P_o - P_e}{1 - P_e} \tag{9}$$

where P_o is the observed percentage of agreement and P_e is the expected percentage based on given his-

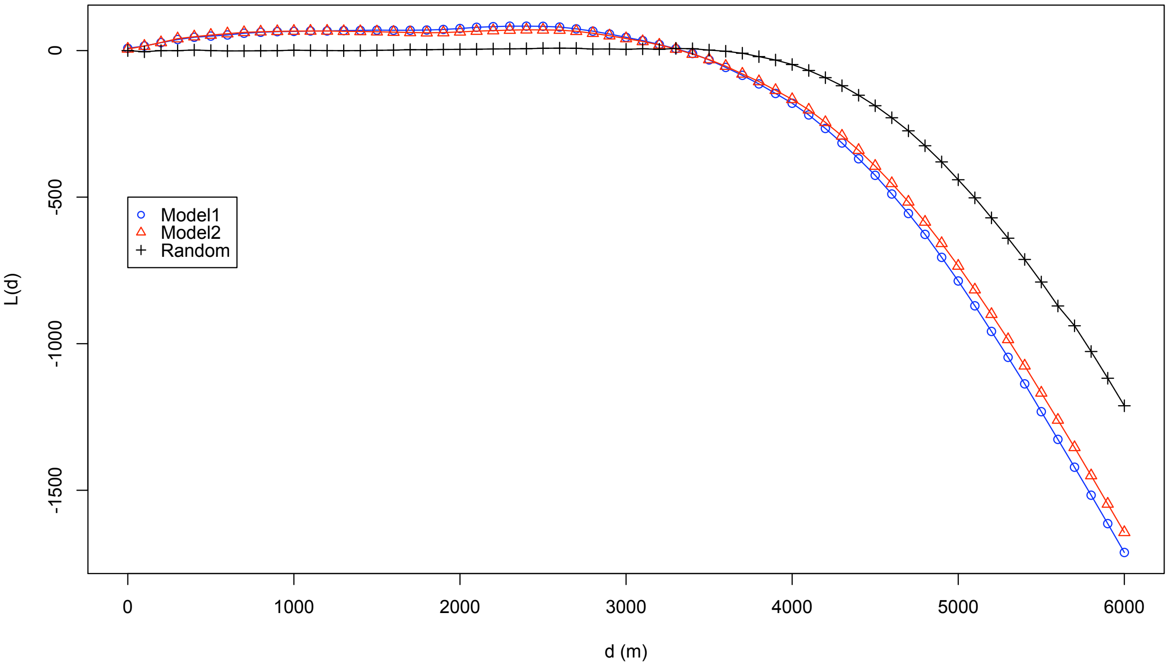

Figure 13: $L(d)$ functions for model and random data sets.

tograms. The above equation is identical to the standard Kappa statistic, except that Hagen derives a fuzzy calculation of P_e. The range of the statistic is from 0 to 1, where 1 represents identical maps.

To use the Fuzzy Kappa statistic, density surfaces were generated for the model data and these were then converted into discrete categories to represent "high", "medium" and "low" crime rates. For this example the author chose thresholds visually which appeared to segment the data approximately equally. If, however, these techniques were used for real analysis a more formal classification method must be used. Then the Map Comparison Kit (Hagen-Zanker et al., 2005) software was used to calculate the Fuzzy Kappa statistic. Figure 14 illustrates the categorical maps and the local Fuzzy Kappa results. The global statistic, $K_{fuzzy} = 0.616$ which indicates some similarity. Interestingly this value is not very high which might be expected given how similar two patterns are. In isolation the global value is not particularly useful but would be more useful if used to compare different results (i.e. to determine whether a new model configuration more closely matches expected data).

The Fuzzy Kappa statistic (and therefore other raster map comparison methods) show promise. However, there is huge scope for variation in the analyses because there are numerous parameters involved in each stage for which numerical values must be assigned. Moreover, the choice of values for these parameters is often arbitrarily. The stages, and places where error can arise, used to calculate Fuzzy Kappa are as follows:

1. **Creating a raster surface**. Using the KDE algorithm, both the size of the kernel and the weighting method can be varied which will significantly influence the density surface produced. This was demonstrated in Section 2.1.2. It is possible to avoid this step by simply aggregating to a regular grid rather than creating a density surface, but as was noted in Section 2.1.2, producing a density surface is a better way

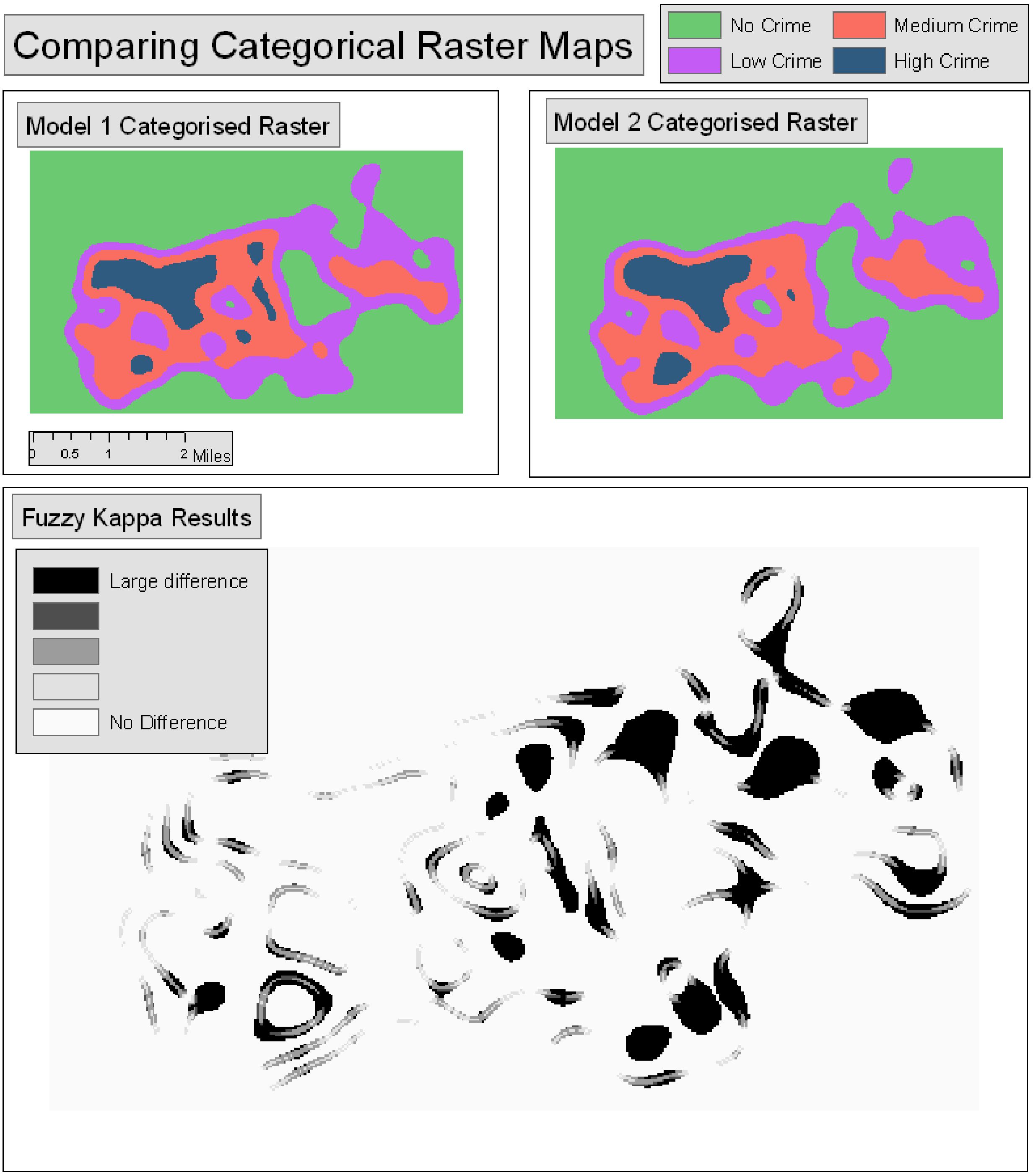

Figure 14: Using the Fuzzy Kappa statistic (Hagen, 2003) to compare Model1 and Model2 data. The overall kappa value is 0.616. The results map illustrates areas of similarity.

of representing crime patterns and also makes it possible to generate neater categorical raster grids (see Figure 3, page 9, for an example of the “messiness” of aggregated cell rasters).

2. **Defining categorical values**. To convert the density surface to a categorical raster map it is necessary to define numerical ranges for the categories. However, numerous threshold methods could be used to set the range values (such as equal interval, equal count, standard deviations etc.) and the number of categories can be varied: both of these will affect the categorical map created.

3. **Raster comparison statistics**. The Fuzzy Kappa statistic itself also uses numerous parameters which will influence how it behaves. These include the size of the neighbourhood, the distance decay function (exponential, linear or constant), parameters within the decay function itself and a definition of the similarity of categories (Hagen-Zanker et al., 2005).

In summary, because there are so many arbitrary parameters it is theoretically possible to completely change the result of the Fully Kappa statistic. Furthermore, as the point patterns themselves are available to analyse it seems unnecessary to lose a great deal of information by converting them to categorical maps. For these reasons, categorical raster map comparison methods will not be used further.

2.2.7 Goodness-of-Fit Statistics

Goodness-of-fit (GoF) statistics are used to describe how well a model matches a set of observations. Knudsen and Fotheringham (1986) experimented with a number of goodness-of-fit statistics and found the standardised root mean square error (SRMSE) to be the best performing. A drawback with SRMSE, however, is that it is difficult to interpret. An alternative statistic, R^2, solves this problem because it represents the percentage of agreement between the model and the expected data. However, R^2 is insensitive to the overall amount of error, predicting a good fit in some circumstances where the SRMSE would not (Harland, 2008). Therefore, both statistics will be used when reporting model accuracy; SRMSE because it provides a more suitable measure of fit for this application and R^2 because it is easier to interpret.

The SRMSE is defined by Heppenstall (2004) as:

$$SRMSE = \frac{\sqrt{(\sum(y_i^{'} - y_i)^2)/n}}{\bar{y}} \tag{10}$$

where $y_i^{'}$ is the predicted value at matrix point i, y_i is the actual value at i, $\bar{y}$ is the mean value of the predicted values ($y^{'}$) and n is the total number of values. The lower limit of the statistic is 0 which indicates no difference between the predicted values (y'_i) and the observed values (y_i). The upper limit is usually 1 (Knudsen and Fotheringham, 1986) but can be greater, particularly when matrices are sparse (Harland, 2008).

Using the same notation, R^2 can be defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - y_i^{'})^2}{\sum_i (y_i - \bar{y})^2} \tag{11}$$

A value of 1 indicates identical data sets and the lower limit of the statistic is 0. To use GoF statistics, the point patterns must first be aggregated in order to create comparable matrices. It is common to aggregate up to a particular administrative boundary to perform the statistics (e.g. Kongmuang (2006)) and Table 2 illustrates the GoF values for the test datasets at different resolutions. The accuracy of the statistics vary considerably depending on the administrative boundaries used but both statistics reveal more similar data at larger resolutions (which is to be expected).

Table 2: Goodness-of-Fit at different resolutions.

Administrative Boundary	Model1 / Model2		Model1 / Random	
	SRMSE	R^2	SRMSE	R^2
Output Area	0.52	0.64	0.98	0.24
Lower Super Output Area	0.23	0.88	0.59	0.47
Medium Super Output Area	0.12	0.98	0.35	0.83

The main problem with this approach, as discussed in Section 2.1.1, is that the process of aggregating to administrative areas leads to the modifiable areal unit problem and the ecological fallacy, whereas aggregating to a regular grid reduces these problems. Therefore, as an alternative to aggregating to administrative boundaries, a new approach is proposed based on that defined by Costanza (1989). As this technique will form the bulk of the model validation it is discussed in greater detail in the following section.

2.3 Expanding cell validation method

As an alternative to aggregating a point pattern up to administrative boundaries, the expanding cell method is an adaption of a technique proposed by Costanza (1989) and involves aggregating points up to a regular grid. This is a more accurate way of representing crime patterns (Chainey, 2009a). GoF statistics can then be applied these grids, by comparing the counts of cells from one pattern to another. However, determining an appropriate resolution for the grid is non-trivial and will significantly effect the GoF values reported. Therefore the results for a range of resolutions will be used. This approach has the advantage of providing information about the scale of accuracy of a model and reducing the effects of the ecological fallacy.

Along with generating reliable GoF values from well-known statistics, the expanding cell method is also able to produce *spatial* estimates for error. Once the counts of points within each cell have been calculated, it is possible to derive the error associated with each individual cell and map this. Various definitions of "error" can be used, but it is important to recognise that in some cases the total counts of points might not be equal (the model might not simulate the total number of burglaries exactly) so proportional values should be used. Here, the relative percentage error between the cells is appropriate because it is easy to interpret and accounts for variable numbers of observations. The relative percentage error between two cells, y_i and $y_i^{'}$, is defined as the difference between the proportions that the cells contribute to the total observation count:

$$100*(\frac{y_i}{\sum y}) - 100*(\frac{y_i^{'}}{\sum y^{'}}) \tag{12}$$

A major benefit of the method is the ability to shift the square grid which helps to account for the modifiable areal unit problem. This shifting is achieved first generating the smallest grid possible which covers all points in the input data. Then the grid is moved 25% the height of a single in the North and Sough directions, and 25% of a cell width in the East or West directions (five distinct grids are created, including the initial one). To ensure that all points are still within the boundary after shifting, a new row or column of cells must be added to the ends of the grid after it has been shifted. Figure 15 illustrates the results of process, comparing the Model1 dataset to some expected results (these data are presented in Section 4). Although there are some differences between the results, in general each grid is able to pinpoint the areas where the data are very different. This highlights the benefit of being able to shift grids with respect to the MAUP: in the original grids there are cells that appears to illustrate a large error, but grid shifting indicates that they are usually anomalies. A further benefit of shifting grids is that the results from all grids can be overlaid to give a comprehensive, "fuzzy" interpretation of the difference between the two input data sets. This method of summarising the results appears to be the most concise and the most informative and will be used when examining model results in later sections.

Other than the ability to display errors spatially, graphs of the global errors (R^2 and SRMSE) also reveal useful information about how well a model is performing. There are various parameters which could be used on the x axis (such as cell length or the total number of cells in the grid) but the most appropriate here is the square area of the cells. This will allow us to investigate the spatial scale that the model is able to make accurate predictions to.

Using trial-and-error, the best curves to fit the R^2 and SRMSE results were found to be:

$$y = \frac{ab + cx^d}{b + x^d} \tag{13}$$

and

$$y = \frac{1}{a + bx^c} \tag{14}$$

respectively, where $a - d$ are constants. Figure 16 illustrates the SRMSE and R^2 errors that arise when comparing both model to expected data and random to expected data. In both cases, as the cell size increases (up to the point that a single cell covers the entire point pattern) the errors decrease. This is to be expected. The graphs also show that at all but the smallest cell sizes the model data are a better approximation of the expected data than the random data set. This is also to be expected but nevertheless comforting that the model is a better predictor of crime than a random process!

2.4 Summary - methods used to validate models

This section has experimented with a number of different methods for both summarising and comparing point patterns. It was decided that the the most reliable method of summarising the a point pattern is to produce a map of the density of the points using the kernel density estimation (KDE) algorithm. To compare point patterns there are two methods that show promise. A graph of the L functions of two point patterns can be compared to determine how similar the clustering of the patterns is and the expanding cell method can be used to produce maps and graphs describing the difference between two point patterns at different resolutions. These methods will be used in subsequent sections to compare results from different model configurations and to expected data.

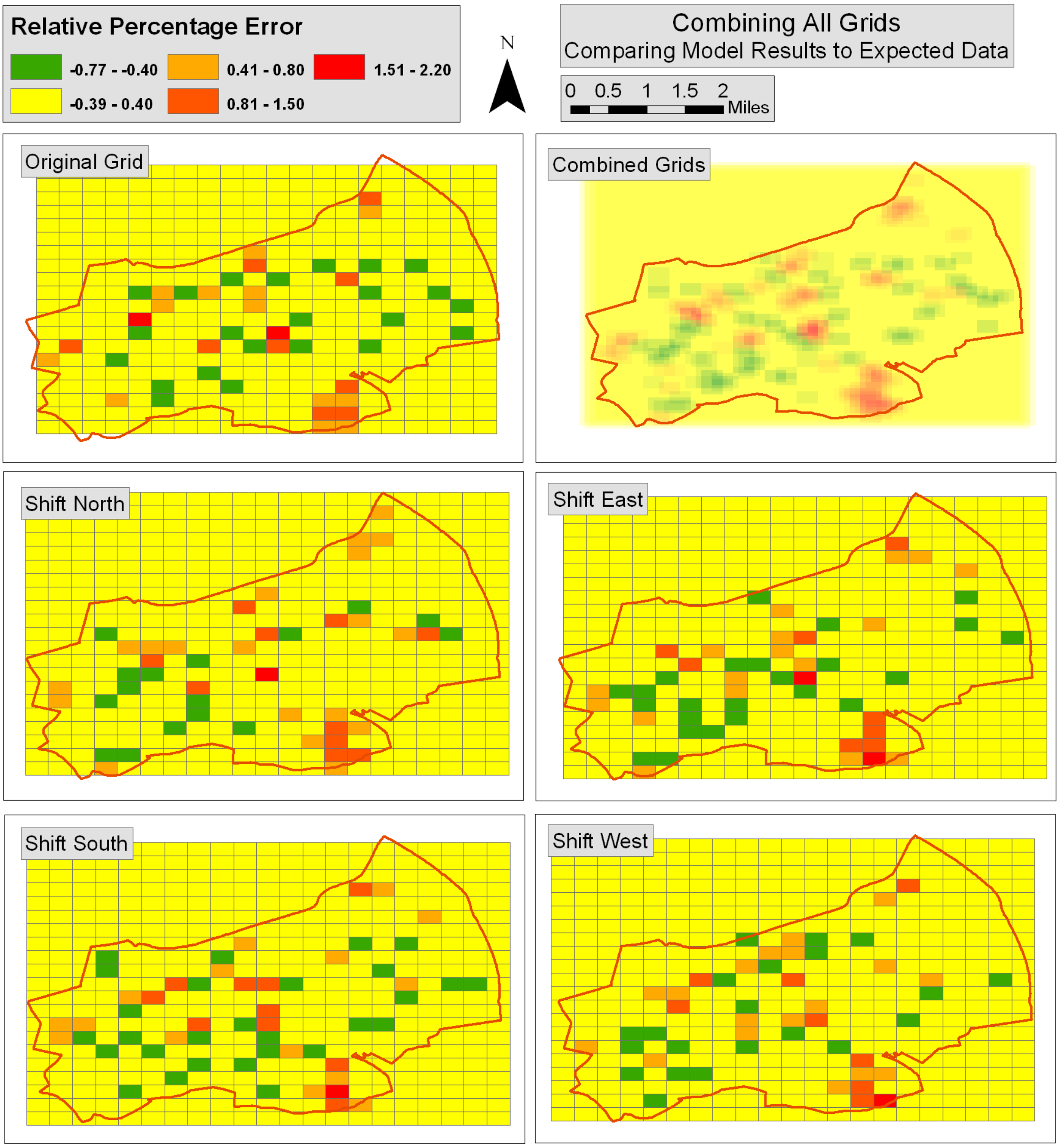

Figure 15: Effects of shifting the expanding cell grid comparing model results to expected data (0.180km^2 cell area).

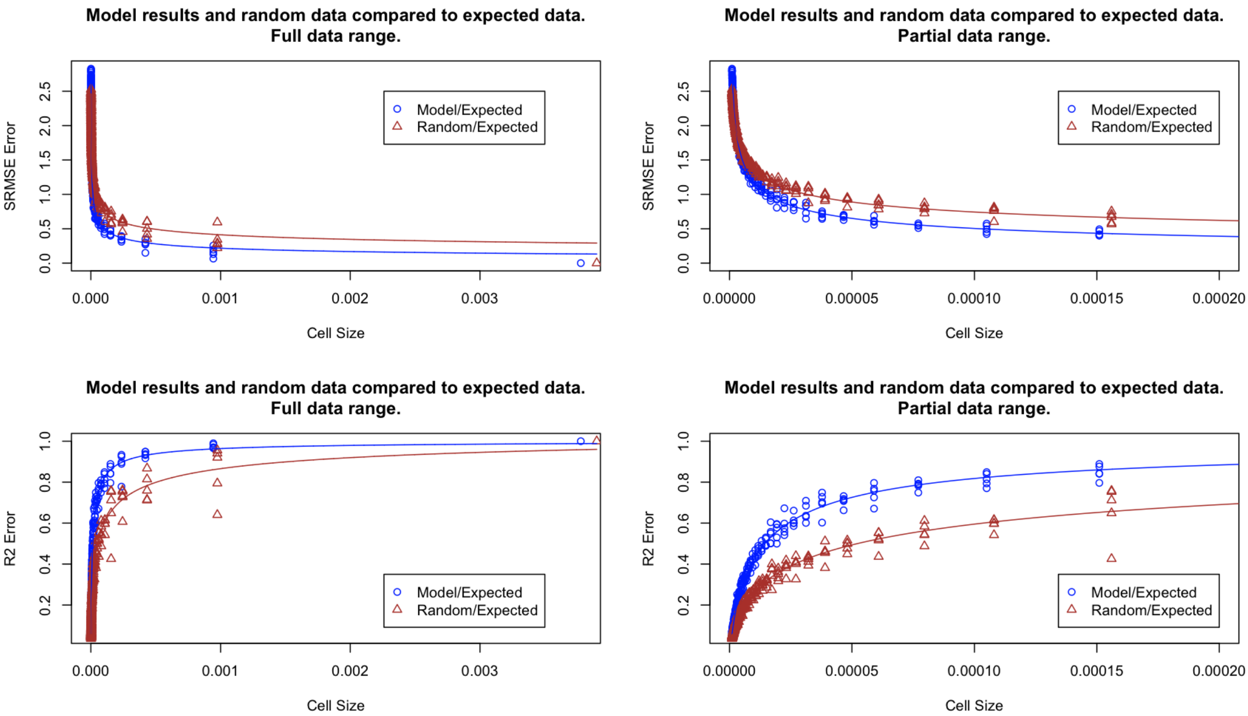

Figure 16: SRMSE and R^2 errors comparing model data and random data to expected data.

3 Verification and Sensitivity Testing

Having determined how to compare model results to each other and to expected data it is now possible to begin evaluating the model. As referred to in Section 1.1, the first stage of the process is termed "verification" and will be conducted by applying the same model to different types of environment to check that the model is performing as it is expected to and, if not, thereby revealing any errors. At the same time, the sensitivity of the model to different parameters will be tested to further our understanding of the dynamics of the model and to ensure that all parameters are affecting the model as they should. Ideally, every parameter would be tested but this model is highly complicated and includes a very large number of parameters. In the absence of sufficient resources to systematically test *every* parameter, therefore, if a parameter is known to have little effect over the model it will not be tested directly.

In order to isolate the effects of a particular variable from the complexities of a highly detailed virtual environment, three different types of environment have been implemented which can be "plugged-in" seamlessly without any changes to the underlying model. The following environments will be used:

1. An **a-spatial** or "**null**" environment in which every journey time can be specified exactly, regardless of the places that the agent is travelling from and to. For example, if an agent needs to travel to a house to purchase drugs the environment simply monitors the journey time.

2. A simple Euclidean **grid** environment which is designed to loosely represent part of a real city. Running the model on this type of environment offers fine-grained control over the geographical locations of different buildings and community types and had the added advantage of considerably shorter execution times (because agent routing is considerably simpler).

3. A complex **GIS** environment which accurately represents a real city. This is the most detailed and accurate environment type and will be used to make real-world crime predictions.

Table 3 documents each model parameter, how it affects the model and which environment will be used to test it. The criteria used to determine which environment should be used to test the parameter is based on the principle of being able to isolate the variable from as much extra complexity as possible. Therefore if the variable can be tested in the "spaceless" null environment it will be, otherwise it must be tested using the grid. Furthermore, the effects of some spatial variables can be subtle and wide ranging; in these cases the parameters require the size and variability provided in a GIS space. Groups of variables in Table 3 are segregated into different categories depending on whether they affect the burglars (i.e. behaviour parameters), the environment (i.e. house, road and community parameters) or other parts of the model (i.e. the starting locations of the burglars). Variables in parentheses are included here for clarity but do not need to be tested directly either because they are dependant on other factors (such as an agent's level of wealth) or because they are not expected to have a large effect on the model (if this is the case it will be justified).

Table 3: All model parameters and the environment which will be used to test the sensitivity of the model to their values.

Name	Description
AGENT VARIABLES	
PECS Agent Parameters. Internal agent parameters can be isolated from space and tested in the **NULL** Environment	
(Wealth)	The amount of wealth an agent has, required to socialise or buy drugs.
Work Gain	The amount of wealth gained per iteration of working.
(Sleep)	The level of sleep an agent has.
SleepGain	Amount of sleep gained per iteration when sleeping (configured by default so that agents want to sleep for 8 hours per day).
(Social)	The agent's current level of the social variable.
SocialGain	Amount social gained per iteration of socialising (configured by default so that agents want to socialise for 2 hours per day).
CostSocialise	The cost of socialising per iteration.
(Drugs)	The level of drugs in the agent.
DrugsGain	Level of drug increase gained by taking drugs (configured by default so that agents want to take drugs once per day).
CostDrugs	Amount of wealth required to buy drugs.
DeteriorateAmount	The rate that internal state variables (e.g. sleep, drugs) will decrease.
Burglary Agent Parameters. Affect the agents' decision regardless of spatial location of the buildings: **NULL** Environment	
(BurgleGain)	The amount of wealth generated by a burglary. This parameter is not tested because it will simply dictate how often an agent needs to burgle, an effect that can be produced by varying other parameters (such as the cost of drugs for example).
Target weights	The weights that affect the area in which the agent will start searching for a burglary target. Based on the distance to the area, its attractiveness, the similarity to the agent's home community and the number of previous successes.
Victim weights	The weights which affect whether or not the agent will burgle a specific house. Based on the accessibility, visibility and security of the house, the traffic volume of the nearest road and the collective efficacy and occupancy levels of the community the house is in.
(SearchTime)	The amount of time to spend searching before choosing a new target to travel to. This is unlikely to have a large effect and will not be tested.
Spatial Agent Parameters. Inherently spatial so will be tested in the **GRID** Environment	
Home locations	The locations of agents' homes.
Social locations	Locations that agents go to in order to socialise.
Work locations	Locations that agents go to in order to work.
Drug dealer locations	Locations of drug dealers.
NumBurglars	The number of burglars in the model.

Name	Description
(Cognitive map)	The buildings and communities which the agent knows about and the number of crimes they have committed there.
ENVIRONMENT VARIABLES	
Building Parameters. Internal building parameters can be isolated from space and tested in the **NULL** Environment	
Accessibility	How easy a house is to break into. Based on the number of possible entrances (i.e. doors or windows).
Visibility	How visible the house is to neighbours or passers-by. Based on the size of the garden and the degree of isolation (the number of neighbours within a given buffer region).
Security	Security level of the house. No empirical data to set to a definite value, but included to increase scope of the model for future work.
TrafficVolume	The amount of traffic outside a house, based on the space syntax integration analysis of the road the house is nearest to.
Community Parameters. Internal community parameters can be isolated from space and tested in the **NULL** Environment	
CollectiveEfficacy	The level of community cohesion.
Sociotype	The Output Area Classification (OAC) (Vickers and Rees, 2006) supergroup of the community.
Attractiveness	The general level of attractiveness of the community as calculated from OAC variables.
Occupancy	The expected levels of occupancy of the community at different times of day as calculated from OAC variables and depending on the time of day.
(OccupancyWeights)	The weights applied to different OAC variables to determine the level of occupancy of different communities.
Burglary Environment Parameters. Inherently spatial because security increases depend on the locations of the buildings in relation to each other. These variables are used during the calibration phase (Section 4) so are not tested here.	
(SecurityIncrease)	The amount that security will increase in a house following a burglary
(SecurityDistance)	The amount that security will increase after a burglary for properties 1 unit[2] away from the burglary (this deteriorates so that further properties are also affected but not by as much).
(SecurityDeteriorate)	The amount that security will deteriorate per day before returning to its base level.
OTHER PARAMETERS	
Other Spatial Variables. Require a relatively large and varied environment so will be tested in a **GIS** Environment.	

[2] A "unit" of length here is equal is the length of a cell in the grid environment or YYYY meters in the GIS environment

Name	Description
(IterPerDay)	The number of iterations per day. This will affect the number of properties that the agent will pass at each iteration of travelling. This is included for flexibility (to find a balance between realism and computational complexity) but is unlikely to have a significant impact so will not be tested.
Transport availability	The different types of transport available to the agent (i.e. buses, car, train etc.).
(Transport Speeds)	The speed gains provided by different transportation methods: bus, train, car on a normal road and car on a major road. These are applicable only to the Vancouver case studies at this stage but can be readily included for Leeds as future research.
Other Non-Spatial Variables. These parameters can be isolated from space and therefore tested in a **NULL** Environment.	
ConstTravelTime	The constant value used in the NULL environment to determine how many iterations it will take for an agent to reach their destination. Testing this value in the NULL environment will provide an indication of the maximum distances that agents are able to travel before they are unable to satisfy their motives in spatial environments. This variable does not apply to other environments, where travel time is determined by geography.
(TravelPerTurn)	The distance the agent can travel when walking. Currently 100m/min (4mi/h) in GIS or 1 unit in grid. This parameter is not tested because the distances that agents can travel each turn are experimented with by varying the effects of transportation.

3.1 Tests Using The Null Environment

PECS (short for "Physical conditions, Emotional states, Cognitive capabilities and Social status") is a cognitive framework that is used to control the burglar agents; providing them with a means of performing intelligent goal-directed behaviour. For more information about how the PECS cognitive framework has been used to control agents see Malleson et al. (2009a). For completeness this section provides extensive evidence that the PECS and the burglary parameters have the desired effect on the model. If desired, the reader can reading to observe how the PECS variables are tested or continue to page 38 to discern how burglary variables are tested. Alternatively, Section 3.2 (page 45) provides the first spatial tests of the model in a grid environment and Section 3.3 (page 50) extends the spatial tests to a realistic GIS environment.

There are two categories of tests which will be performed using the Null environment. The first category encompasses all the PECS variables that drive agent behaviour; it must be determined how agents' internal state variables are influenced by external actions. For example, what will happen to an agent's state variables if the cost of socialising increases and how will this affect their behaviour? The second category relates to burglary; how will external building or community parameters and internal agent burglary parameters influence where an agent chooses to burgle? The simulation will contain only a single agent for clarity and because agents themselves do not interact directly so will not influence each other in these tests. The environment will be configured slightly differently for each category of test. For the first category (PECS variables) there will only be a single building of each type (a house, work place, drug dealer and social place) to limit the complexity. All buildings will be contained in the same community and will have default parameter values for all of their internal variables (0.5). This is illustrated by Figure 17. In the second category (burglary variables) there will be different numbers of houses and communities in order to compare the effects of changing house and community parameter values (this is discussed in Section 3.1).

With regards to timing, a single iteration is defined as 1 minute of virtual time; this equates to $60 * 24 = 1440$ iterations in each virtual day. This is sufficient to provide a high temporal resolution and has the elegance of linking simulated minutes to iterations directly. The simulation will be run for 43,200 iterations (30 days). This was chosen because it is at least sufficient for all the simulations to reach equilibrium (as Section 4 will illustrate).

Although there is no space in the Null environment, agents still *think* that they are travelling between objects. The Null environment, therefore, controls how long each hypothetical journey takes. For these experiments the constant travel time will be set at 30 minutes/iterations but the travel time variable will also be tested in the first experiment with the PECS variables.

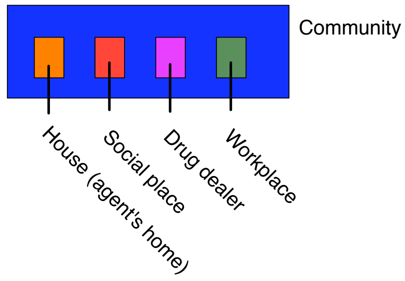

Figure 17: The layout of the environment to test PECS variables.

Testing PECS Variables

The following describes how the parameters have been configured initially. These follow decisions that were based on the initial experimentation with a prototype model. Assumptions will be justified afterwards.

- State variable levels deteriorate at a constant rate of 1 unit per day. The approximate amount of time that agents must spend increasing their state variables can be altered by changing the return gained by working, sleeping, socialising or taking drugs. For example, decreasing the amount of sleep gained from one hour of the "sleep" action effectively increases the amount of time the agents must spend sleeping.
- With default gains from the appropriate actions, agents want to spend approximately 8 hours sleeping, 2 hours socialising and want to take drugs once per day to maintain comfortable state variable levels.
- Working for 8 hours or committing a single burglary provides 1 unit of wealth. Socialising for two hours costs 0.25 units and taking drugs costs 0.5 units. Therefore agents will need to work for approximately 6–8 hours per day or commit slightly less than one burglary per day on average to make enough wealth to satisfy their needs for socialising and drug addictions.

These parameters are global at this stage, making agents homogeneous. However, it is trivial to vary parameters for individual agents which will lead to a heterogeneous population and allow researchers to experiment with different burglar behaviour and motivations – this is one of the major benefits of agent-based modelling (Axtell, 2000; Castle and Crooks, 2006). With regards to the assumptions made for these parameter values, they have initially been chosen because they will simulate, approximately, the daily patterns that might be exhibited by people employed in typical "9–5" jobs. The largest effect that the variable values will have on the model will be to develop the burglars' awareness spaces and subsequently influence where they will look for burglary targets. For example, it is important that the need to socialise expands an agent's awareness space to parts of the city that they would otherwise not explore. But it is less important whether or not they spend one or two hours actually socialising when they get there. Therefore, although the effect that the variables have on the model will be thoroughly tested here, seeking additional evidence to help choose their initial values is not necessary.

Table 4 outlines the variables that will be tested and their default values (that are set to reflect the previous assumptions based on the number of iterations per day). Using the gain from work as an example: if agents should spend 8 hours working and there are 1440 iterations per day, or $(1440/24) * 8 = 480$ iterations in an 8 hour period, then each iteration must provide $1/480 = 0.002083$ units of wealth per iteration. If the number of iterations per day changes these values must obviously be updated to reflect these changes. The ConstTravelTime variable is the time that each journey will take in the null environment and is also tested here to determine the maximum length of journeys before the agents are unable to satisfy their motives. Its initial value is set to 30 minutes.

Table 4: Default values and range for PECS variables to be tested.

Variable	Initial Value
WorkGain	0.002083
SleepGain	0.002083
SocialGain	0.008333
DrugsGain	1
CostSocialise	0.002083
CostDrugs	0.5
ConstTravelTime	30
DeteriorateAmount	0.000694

A *state variable* is a PECS variable that represents the internal level of some need. For example, the *sleep* state variable represents an agent's "sleepiness". The state variable has an associated *motive* which competes with other motives to drive the agent's behaviour (the strongest motive wins). So with sleeping, as the *sleep* state variable approaches zero the associated motive becomes very large and will take control of the agent who will subsequently go home to sleep. If a state variable value reaches zero, however, the model cannot function as there is no way to calculate the associated motive intensity. The model becomes inoperative. This occurrence could represent a significant turning point for the agent which causes a dramatic change in their behaviour (such as choosing to break a drug habit) but exploring this is beyond the scope of the research. Instead, it is important to establish the range of parameter values that will cause a state variable to drop to zero as this is the effective minimum/maximum value of the parameter and therefore indicates how sensitive the model is to the parameter.

Figure 18 graphs the values of all state variables and motives for a single burglar when each parameter has been set to the initial value outlined in Table 4. Only iterations 0 – 5,000 are shown for clarity, although the same pattern is repeated for the duration of the simulation. After an initial period of disturbance the agent forms a routine and both state variable levels and motive intensities stabilise. All state variables deteriorate over time unless the agent is performing an activity which increases their value. The state variable levels for sleep and socialising increase steadily as the agent sleeps or socialises respectively. The drugs state variable, on the other hand, increases instantaneously when the agent takes a "hit". The pattern for motive intensities is less orderly because the intensity is driven by the time of day as well as the size of its associated state variable, but motive intensities also reach equilibrium nonetheless.

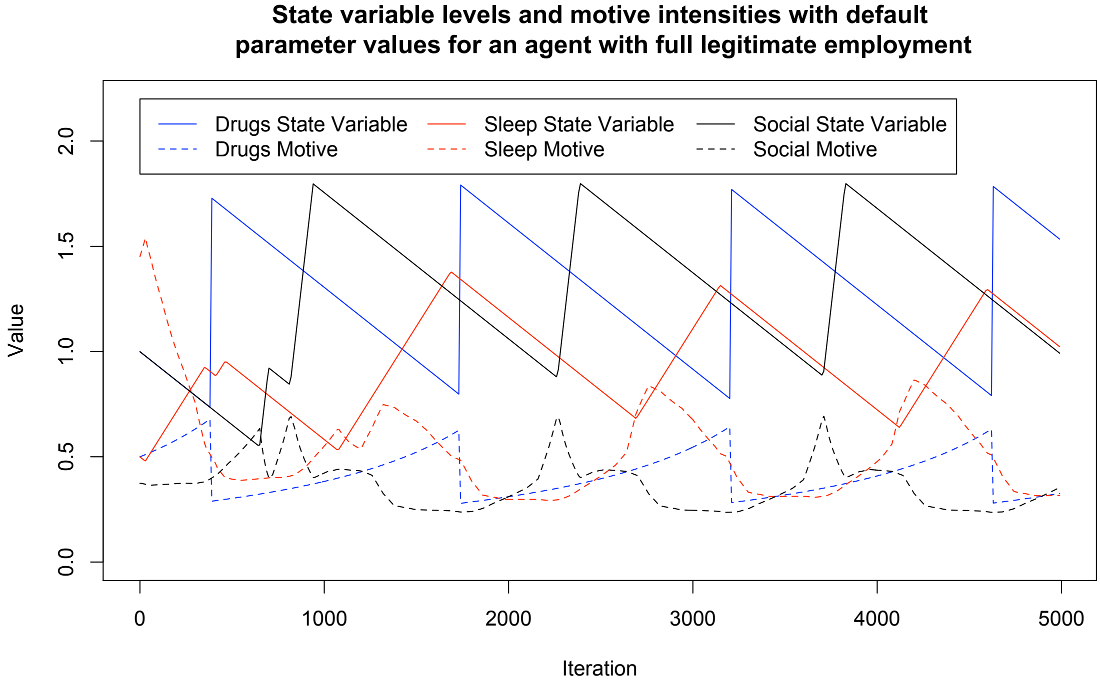

Figure 18: The intensity of motives and values of state variables for a single agent under initial parameter values. The agent has full-time employment.

For completeness, the same simulation was executed again but this time with an agent who had no temporary employment and was forced to burgle to gain wealth. The full results are illustrated in Figure 19. The simulation terminates after approximately 10,000 iterations as a state variable value reaches 0, as evidenced by the dramatic increases in motive intensities shortly before the simulation terminates. The reason for the model failure is is because there is only a single house in the environment and the default behaviour is for it to increase its security increases after each burglary. Therefore after a number of burglaries the agent is no longer able to burgle there any more and cannot generate the wealth required to satisfy its needs. Although the effects of burglary will be investigated further shortly, this result nevertheless illustrates that the model is performing as expected.

This proof-of-concept experiment establishes that the simulation is performing as expected under default conditions and in the absence of a geographical space, it is possible to begin to vary the values of parameters to establish what affect they will have on the model. Table 5 illustrates the results of varying the WealthGain parameter which represents the amount of wealth given, per iteration, to agents who are working. It should be noted that the parameter was varied in regular intervals between zero and ten times its default value but, for clarity, only some results are included in the table (these are representative of the pattern shown by all).

The first two simulations fail: agents are not able to generate enough wealth to satisfy their needs and the simulations terminate after 2,880 and 10,344 iterations respectively as shown by the NumIter field (the maximum number of iterations reached). The remaining simulations illustrate firstly that the amount of time agents spend working ("Avg Work Time") reduces significantly as the gain from work increases. Furthermore, the mean levels of state variables are generally larger and motive intensities generally lower as the return from working increases. This is to be expected because the agents do not need to spend as long working and can,

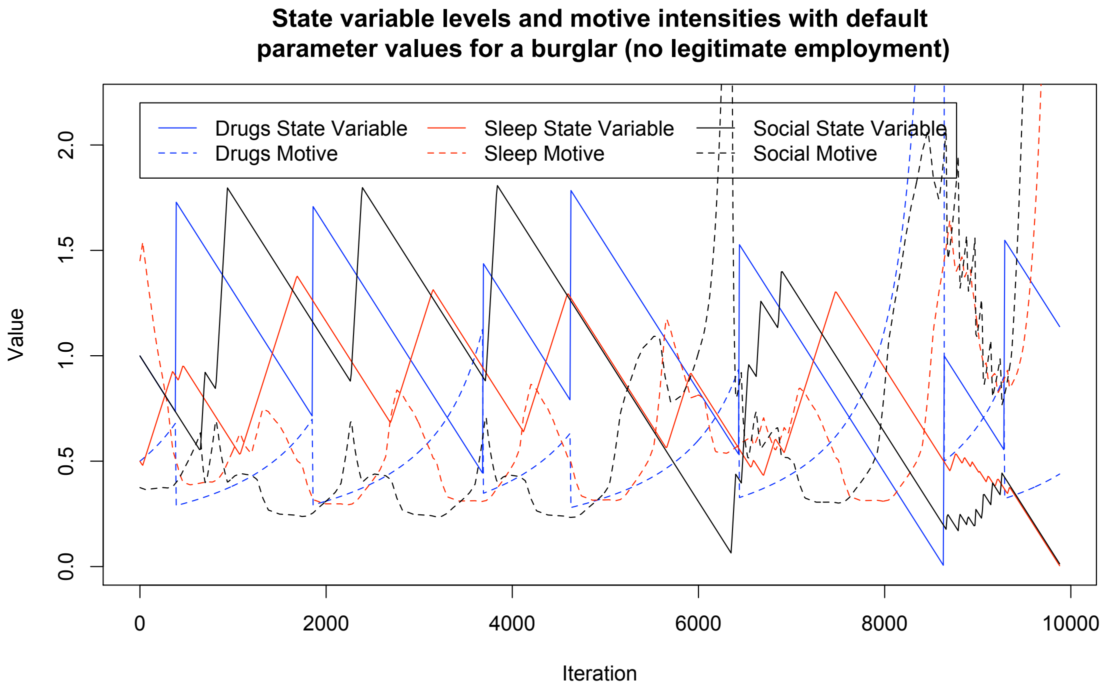

Figure 19: The intensity of motives and values of state variables for a single agent under initial parameter values. The agent must burgle to generate wealth.

Table 5: Summary results for tests of the WealthGain parameter. As the parameter increases, the agent can spend less time working.

WorkGain value	0.0	0.001	0.002	0.01	0.02	0.02083
Max Iteration	2880	10344	43200	43200	43200	43200
Avg Sleep SV	-	-	0.99	0.95	0.95	0.95
Avg Sleep M	-	-	0.48	0.51	0.51	0.51
Avg Social SV	-	-	1.22	1.41	1.4	1.4
Avg Social M	-	-	0.41	0.34	0.34	0.34
Avg Drugs SV	-	-	1.14	1.23	1.24	1.24
Avg Drugs M	-	-	0.47	0.43	0.43	0.43
Avg Travel Time	-	-	0.15	0.15	0.15	0.15
Avg Work Time	-	-	0.25	0.05	0.02	0.02
Avg Social Time	-	-	0.09	0.09	0.09	0.09
Avg Sleep Time	-	-	0.34	0.34	0.34	0.34
Avg DoNothing Time	-	-	0.18	0.38	0.4	0.4
Avg Drug Take	-	-	30	30	30	30

instead, satisfy other motives. This is evident by the increases in time spent doing nothing ("Avg DoNothing Time") where there is no motive strong enough to control the agent.

The remainder of this section will outline the effects of varying other parameters. It should be noted that only interesting results are included so that the values for test variables provided are not necessarily consistent. The results for varying SleepGain and SocialGain values are similar to WealthGain as would be expected, affecting the amount of time that agents spend sleeping and socialising respectively. These are illustrated in Table 6 and Table 7. Varying DrugsGain results are illustrated in Table 8. The simulation fails if the value is less than 0.5 and after that the number of times the burglar must take drugs decreases. Again more time is spent doing nothing as the agent does not need to make money to buy drugs as often.

Table 6: Summary results for tests of the SleepGain parameter. As the parameter increases the agent can spend less time sleeping.

SleepGain Value	0.0	0.001	0.002	0.01	0.02	0.02
NumIter	721	4812	43200	43200	43200	43200
AvgSleepV	-	-	0.98	1.23	1.54	1.49
AvgSleepM	-	-	0.49	0.38	0.32	0.33
AvgSocialV	-	-	1.22	1.22	1.22	1.21
AvgSocialM	-	-	0.41	0.41	0.41	0.41
AvgDrugsV	-	-	1.14	1.2	1.23	1.23
AvgDrugsM	-	-	0.47	0.45	0.43	0.43
AvgTravelTime	-	-	0.15	0.17	0.19	0.18
AvgWorkTime	-	-	0.24	0.24	0.24	0.24
AvgSocialTime	-	-	0.09	0.09	0.09	0.09
AvgSleepTime	-	-	0.35	0.07	0.04	0.03
AvgDoNothingTime	-	-	0.17	0.43	0.45	0.46
AvgDrugTake	-	-	30	30	30	30

Table 7: Summary results for tests of the SocialGain parameter. As the parameter increases the agent can spend less time socialising.

Social Gain Value	0.000	0.002	0.004	0.005	0.010	0.050	0.083
NumIter	1441	3603	15126	43200	43200	43200	43200
AvgSleepV	-	-	-	0.91	0.95	0.95	0.95
AvgSleepM	-	-	-	0.54	0.51	0.51	0.51
AvgSocialV	-	-	-	0.87	1.37	1.37	1.36
AvgSocialM	-	-	-	0.57	0.36	0.36	0.37
AvgDrugsV	-	-	-	1.14	1.11	1.17	1.17
AvgDrugsM	-	-	-	0.47	0.48	0.46	0.46
AvgTravelTime	-	-	-	0.17	0.15	0.15	0.15
AvgWorkTime	-	-	-	0.3	0.23	0.17	0.17
AvgSocialTime	-	-	-	0.14	0.07	0.01	0.01
AvgSleepTime	-	-	-	0.34	0.34	0.34	0.34
AvgDoNothingTime	-	-	-	0.06	0.22	0.33	0.34
AvgDrugTake	-	-	-	30	30	30	30

Table 8: Summary results for tests of the DrugsGain parameter. As the gain from drugs increases the agent requires fewer "hits" (the AvgDrugTake field).

Drugs Gain Value	0	0.1	0.2	0.3	0.4	0.5	1.00	5.00	11.00
NumIter	1441	2304	3156	4032	7416	43200	43200	43200	43200
AvgSleepV	-	-	-	-	-	0.93	0.99	1.12	1.12
AvgSleepM	-	-	-	-	-	0.52	0.48	0.44	0.44
AvgSocialV	-	-	-	-	-	1.01	1.22	1.35	1.35
AvgSocialM	-	-	-	-	-	0.47	0.41	0.36	0.36
AvgDrugsV	-	-	-	-	-	0.87	1.15	3.17	6.56
AvgDrugsM	-	-	-	-	-	0.59	0.47	0.21	0.12
AvgTravelTime	-	-	-	-	-	0.17	0.15	0.13	0.12
AvgBurgleTime	-	-	-	-	-	0	0	0	0
AvgWorkTime	-	-	-	-	-	0.41	0.24	0.11	0.09
AvgSocialTime	-	-	-	-	-	0.08	0.09	0.09	0.09
AvgSleepTime	-	-	-	-	-	0.34	0.34	0.34	0.34
AvgDoNothingTime	-	-	-	-	-	0	0.19	0.34	0.36
AvgDrugTake	-	-	-	-	-	60	30	6	3

Tables 9 and 10 illustrate the effects of increasing the cost of socialising and taking drugs. In both cases the agent must spend more time working as the variable values increase. Once the variables become too large the simulation breaks down as the agent is not able to generate sufficient wealth to satisfy their needs.

Table 9: Summary results for tests of the CostSocialise parameter. Once the cost of socialising becomes too great the simulation fails as agents are not able to fulfil the need to socialise.

CostSocialise Value	0.000	0.003	0.005	0.006	0.007	0.008	0.009
NumIter	43200	43200	43200	17604	7776	6324	4896
AvgSleepV	0.98	0.94	0.9	-	-	-	-
AvgSleepM	0.49	0.52	0.54	-	-	-	-
AvgSocialV	1.36	0.96	0.84	-	-	-	-
AvgSocialM	0.37	0.52	0.58	-	-	-	-
AvgDrugsV	1.19	1.14	1.11	-	-	-	-
AvgDrugsM	0.45	0.47	0.49	-	-	-	-
AvgTravelTime	0.14	0.17	0.19	-	-	-	-
AvgWorkTime	0.16	0.28	0.36	-	-	-	-
AvgSocialTime	0.09	0.08	0.08	-	-	-	-
AvgSleepTime	0.34	0.34	0.34	-	-	-	-
AvgDoNothingTime	0.28	0.13	0.03	-	-	-	-
AvgDrugTake	30	30	30	-	-	-	-

Varying the amount of time each simulated journey takes to complete (the ConstTravelTime variable) directly influences how much time the agent is able to spend doing nothing, as illustrated by Table 11. The amount of time doing nothing decreases while travel time increases because the agent must spend more time travelling around the environment. At approximately ConstTravelTime ≈ 100 the agent is not able to satisfy their needs because each journey takes such a considerable amount of time and the simulation breaks. This information

Table 10: Summary results for tests of the CostDrugs parameter. Once the cost of drugs is too high the simulation fails because the agent is not able to generate enough money to satisfy their drug addiction.

CostDrugs Value	0	0.25	0.5	0.75	1	1.25	1.5	1.75	2
NumIter	43200	43200	43200	43200	43200	8640	4596	3642	2880
AvgSleepV	1.02	1	0.99	0.97	0.95	-	-	-	-
AvgSleepM	0.47	0.48	0.48	0.5	0.51	-	-	-	-
AvgSocialV	1.35	1.23	1.22	1.12	1.06	-	-	-	-
AvgSocialM	0.36	0.41	0.41	0.45	0.48	-	-	-	-
AvgDrugsV	1.29	1.23	1.15	1.09	1.02	-	-	-	-
AvgDrugsM	0.41	0.43	0.47	0.5	0.54	-	-	-	-
AvgTravelTime	0.14	0.15	0.15	0.15	0.15	-	-	-	-
AvgWorkTime	0.08	0.16	0.24	0.33	0.41	-	-	-	-
AvgSocialTime	0.09	0.09	0.09	0.08	0.08	-	-	-	-
AvgSleepTime	0.34	0.34	0.34	0.34	0.34	-	-	-	-
AvgDoNothingTime	0.36	0.27	0.19	0.1	0.02	-	-	-	-
AvgDrugTake	30	30	30	30	30	-	-	-	-

will be useful when configuring the speed at which the agents are able travel around the environment.

Table 11: Summary results for tests of the ConstTravelTime parameter.

Const Travel Time Value	0	25	50	75	95	100	110	300
NumIter	43200	43200	43200	43200	43200	13356	6036	1299
AvgSleepV	1.04	1	0.89	0.85	0.83	-	-	-
AvgSleepM	0.46	0.48	0.55	0.58	0.6	-	-	-
AvgSocialV	1.38	1.22	0.92	0.93	0.88	-	-	-
AvgSocialM	0.35	0.41	0.56	0.56	0.63	-	-	-
AvgDrugsV	1.23	1.17	1.13	1.05	0.99	-	-	-
AvgDrugsM	0.43	0.46	0.48	0.52	0.56	-	-	-
AvgTravelTime	0	0.12	0.24	0.33	0.34	-	-	-
AvgWorkTime	0.24	0.24	0.24	0.24	0.24	-	-	-
AvgSocialTime	0.09	0.09	0.08	0.08	0.08	-	-	-
AvgSleepTime	0.34	0.34	0.34	0.34	0.34	-	-	-
AvgDoNothingTime	0.33	0.21	0.09	0.01	0	-	-	-
AvgDrugTake	30	30	30	30	30	-	-	-

Finally, Tables 12 and 13 illustrate the affect of changing DeteriorateAmount and DoNothingThreshold respectively. As the rate that state variables deteriorate (DeteriorateAmount) the amount of time the agents spend doing nothing decreases until they are no longer able to satisfy their needs and the simulation fails. The DoNothingThreshold parameter dictates at what point no motive is intense enough to take control of the agent, allowing the DoNothing motive to take control. Results are as should be expected; agents spend more time doing nothing as the parameter increases up to a point that the simulation breaks (state variables deteriorate to their minimum value before a motive is greater that the DoNothingThreshold). In the cases where the agent spends no time doing nothing all state variables increase in value indefinitely as agents will increase them even if they are already high.

Table 12: Summary results for tests of the DeteriorateAmount parameter.

DeteriorateAmount	0.0000	0.0002	0.0004	0.0006	0.0008	0.0010
NumIter	43200	43200	43200	43200	20802	3384
AvgSleepV	1.68	1.4	1.39	1.17	-	-
AvgSleepM	0.28	0.34	0.35	0.42	-	-
AvgSocialV	1.78	1.42	1.35	1.22	-	-
AvgSocialM	0.27	0.34	0.37	0.41	-	-
AvgDrugsV	1	1.34	1.29	1.19	-	-
AvgDrugsM	0.5	0.39	0.41	0.45	-	-
AvgTravelTime	0	0.06	0.12	0.15	-	-
AvgWorkTime	0	0.07	0.14	0.21	-	-
AvgSocialTime	0	0.03	0.05	0.07	-	-
AvgSleepTime	0.01	0.1	0.2	0.29	-	-
AvgDoNothingTime	0.98	0.75	0.5	0.28	-	-
AvgDrugTake	0	9	17	26	-	-

Table 13: Summary results for tests of the DoNothingThrreshold parameter.

DoNothingThreshold	0	0.25	0.5	0.75	1	1.25	1.5
NumIter	43200	43200	43200	10075	3459	3074	3062
AvgSleepV	5.99	2.16	1	-	-	-	
AvgSleepM	0.13	0.23	0.48	1.1	-	-	-
AvgSocialV	5.28	2.55	1.23	0.49	-	-	-
AvgSocialM	0.12	0.2	0.41	2.18	-	-	-
AvgDrugsV	4.96	2.27	1.16	0.84	-	-	-
AvgDrugsM	0.14	0.23	0.46	0.9	-	-	-
AvgTravelTime	0.13	0.13	0.15	0.31	-	-	-
AvgWorkTime	0.3	0.25	0.24	0.18	-	-	-
AvgSocialTime	0.11	0.09	0.09	0.07	-	-	-
AvgSleepTime	0.46	0.35	0.34	0.32	-	-	-
AvgDoNothingTime	0	0.18	0.19	0.13	-	-	-
AvgDrugTake	36	32	30	6	2	2	2

As a result of these tests it has been established that the PECS behaviours are controlling the agent as they should do in the absence of a spatial model component and without the need to burgle. Furthermore, all parameters have similar affects on the model which is encouraging as it is not the intention of the model design (at this stage) that one variable should have a stronger affect on agents than another. Therefore it is now possible to systematically test burglary parameters in order to examine the effect that they will have on agent behaviour.

Testing Burglary Variables

These tests are important to determine how attractive a house or community is to a burglar (and subsequently whether or not they decide to burgle there) based on the values of relevant parameters. Unlike the previous PECS parameters, the theoretical range of these parameters is 0 – 1 so they will not be driven outside of this range. Table 14 outlines the test variables with descriptions. Parameters are divided into three separate groups: those of buildings; those of communities; and those internal to agents. The agent parameters are denoted with a following "_W" to indicate that they are weights which the burglars apply to an associated house or community variable when contemplating burglary. For example, the *Attractiveness_W* parameter represents how the attractiveness of a community affects an agent when they are deciding where to travel to in order to search for a burglary target; an agent with a high value for the *Attractiveness_W* weight is more likely to travel to a community which is highly attractive and vice versa.

The four parameters from Table 14 in braces cannot be tested in the null environment and will, instead, be tested in the grid environment (Section 3.2). Obviously Distance_W cannot be tested here because there is no space, all distances are the same. The Sociotype and SocialDifference_W variables, on the other hand, cannot be tested here because they cannot be summarised by a single numerical value and also depend on where the agent lives. Therefore it makes more sense to test these in a spatial environment, where different types of community and different agent home addresses can be tested spatially. Also with the PreviousSuccess_W parameter, a relatively large environment is required to determine whether or not the parameter is really influencing where an agent chooses to burgle.

The environment will be configured for the different categories of tests as follows (also illustrated by Figure 20):

- **House parameters**. Here the values of different household parameters are being tested to see how susceptible they are to burglary. There will be a single agent in the environment and a single community, both with default parameter values. To cover a range of parameter values the environment will contain eleven houses, each with an ascending value for the parameter being tested (the first house will have a parameter value of 0.0, the second 0.1 and so on, up to 1.0).

- **Community parameters.** These tests are very similar to the house parameter tests. Again there will be a single agent, but this time the environment will contain eleven communities with ascending test parameter values. Each will contain a single house that will have default parameter values. Again the results can be summarised using a graph of the parameter value against the number of burglaries in the community with that parameter value.

- **Burglar parameters**. Here burglars' perceptions of the environment are being tested; weights are applied to each aspect of a community or building to dictate how an agent perceives the object. To cover a range of values for the weight being tested there will be eleven agents in the simulation, each with an ascending

Table 14: Parameters related to burglary to be tested and a description of their function.

Variable	Description
House Parameters	
Accessibility	The accessibility of the house (how easy it is to break in to).
Visibility	The visibility of the house to neighbours.
Security	The security level of the house.
TrafficVolume	The amount of traffic outside the house.
Community Parameters	
CollectiveEfficacy	The level of collective efficacy (community cohesion).
(Sociotype)	The type of the community.
Attractiveness	The attractiveness (wealth) of the community
Occupancy	The occupancy levels of the community
Burglar Parameters	
TargetWeights:	*Weights that affect how an agent perceives the an associated community variable when planning where to search for a burglary target.*
(Distance_W)	How important the distance to the community is to the agent.
Attractiveness_W	How important community attractiveness is to the agent.
(SocialDifference_W)	How important it is that the community is similar to the burglar's home community.
(PreviousSuccess_W)	How many times the agent has previously burgled in the community.
VictimWeights:	*Weights which affect how an agent perceives a house or community when actually searching for a specific burglary victim.*
CollectiveEfficacy_W TrafficVolume_W Occupancy_W Accessibility_W Visibility_W Security_W	Weights applied to associated house or community parameters.

value for the weight. All other weights will have default values of 0.5. Furthermore, if the weight being tested applies to a house parameter the environment will consist of eleven houses (with ascending related parameter values) and a single community or eleven communities each with a single house if the weight applies to a community parameter. Then the results can be summarised in a table illustrating, for each agent, how many burglaries they committed in each of the houses or communities.

The burglary process incorporates a random component to the agents decision whereby they are more *likely* to burgle if conditions are favourable, but this is not certain. Therefore the robustness of model results must be established by executing the model a number of times. But how many times? Table 15 illustrates the number of burglaries committed in houses with different accessibility levels in ten different model runs. In general, houses which are the most accessible are the most heavily burgled which is to be expected (recall that due to the normalisation process high accessibility houses have values of 0) but there is considerable variation. The model must be executed a sufficient number of times so that expected number of burglaries in a house is approximately equal to the mean of infinitely many executions. Following Voas and Williamson (2000) it is possible to estimate how the standard error of the mean changes with the number of executions. Take, for example, the house with accessibility 0 which appears to receive, on average, 6 burglaries (which is 27% of the

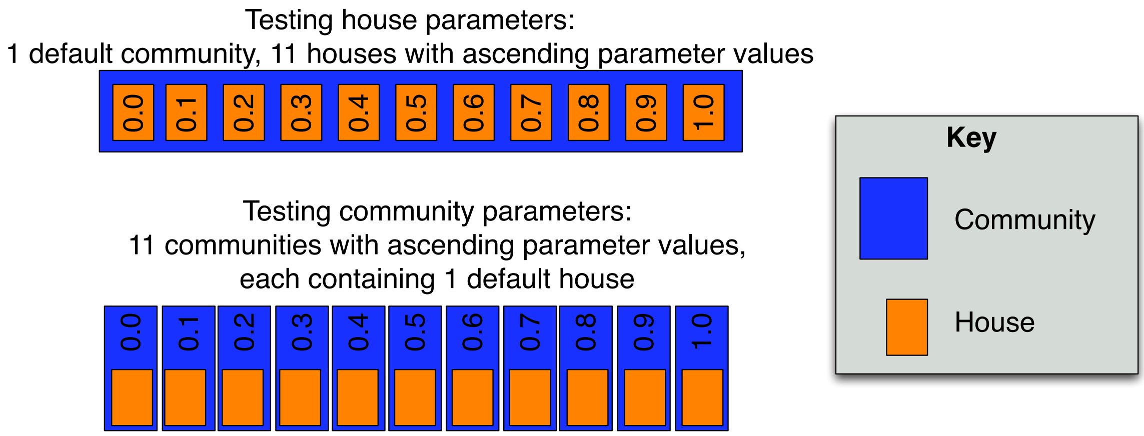

Figure 20: The environmental configurations (including the value of the parameter being tested) when testing house parameters and community parameters using the Null environment.

total of 22 burglaries) with standard deviation 2.0. Over 10 iterations the standard error will be $2.0/\sqrt{50} = 0.28$ which is approximately 11% of 6. Increasing the number of iterations to 20 reduces the standard error to 7%, 50 reduces it to 5% and with 100 iterations the error is approximately 3%. Although these estimates are likely to be inaccurate, they illustrate that the improvements in error fall off as the number of iterations increases. The standard error gains between 50 and 100 iterations are offset by the considerable extra execution times and therefore 50 iterations is deemed sufficient for these tests. Furthermore, the purpose of these tests is to realise *patterns* that the model produces rather than make accurate predictions so it matters little if exact results might change slightly if more iterations were performed, so long as the patterns are accurate.

Table 15: An example of the probabilistic nature of the model: the number of burglaries committed in houses with different accessibility levels in a ten different models (M1 – M10).

House Accessibility	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
0	6	7	8	3	6	7	5	7	3	9
0.1	5	4	1	5	2	3	6	2	8	5
0.2	7	3	4	2	2	5	3	4	4	1
0.3	0	3	0	2	2	3	1	5	2	3
0.4	2	1	3	2	1	1	1	0	1	0
0.5	2	1	5	3	4	1	3	0	1	2
0.6	0	1	1	2	1	2	2	1	2	0
0.7	0	1	0	2	2	0	1	2	0	0
0.8	0	1	0	1	2	0	0	0	0	1
0.9	0	0	0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0	1	0	1

Table 16 illustrates the proportion of burglaries committed in houses or communities with a given parameter value over 50 model runs. Each parameter was tested individually so the table displays the results of seven individual experiments. Most results are as to be expected. The lower the *accessibility* (Acc), *visibility* (Vis), *security* (Sec) and *traffic volume* (TV) house parameters, the more burglaries the houses can expect. Similarly with communities, large values for the *collective efficacy* (CE) and *occupancy* (Occ) parameters lead to more burglaries in the community. The only exception is the *attractiveness* (Att) community variable which does not appear to affect the number of burglaries committed in the community; the proportion is consistent regardless of the parameter value. This is because the attractiveness parameter is used by agents in a different way: instead of directly influencing the likelihood that they will commit a burglary, attractiveness is used to determine where they will *begin to search* for a burglary target. As there are only 11 houses and communities in the environment it is likely that, although the agents start their search in the most attractive community, the final burglary actually occurs elsewhere. In a large spatial environment where each community contains a large number of houses it is more likely that attractiveness will more directly influence where burglaries occur. Sections 3.2 and 3.3 will test this premise.

Table 16: The proportion of burglaries committed in houses or communities with given parameter values over 50 model runs

	House Parameters				Community Parameters		
Parameter Value	Acc	Vis	Sec	TV	CE	Att	Occ
0	0.25	0.27	0.26	0.28	0.26	0.08	0.25
0.1	0.22	0.2	0.23	0.22	0.21	0.09	0.2
0.2	0.16	0.17	0.15	0.14	0.16	0.1	0.17
0.3	0.13	0.12	0.12	0.12	0.13	0.09	0.11
0.4	0.08	0.1	0.09	0.09	0.11	0.09	0.1
0.5	0.07	0.05	0.06	0.06	0.06	0.09	0.06
0.6	0.03	0.04	0.05	0.04	0.04	0.1	0.05
0.7	0.03	0.03	0.03	0.03	0.02	0.09	0.03
0.8	0.02	0.02	0.01	0.02	0.01	0.09	0.01
0.9	0.01	0.01	0	0	0.01	0.1	0.02
1	0.01	0	0.01	0	0	0.08	0

To complete the burglary parameter tests, Tables 17, 18, 19, 20 and 21 illustrate the effects of changing the burglar behaviour parameters. Table 17 shows that burglars with a low Accessibility_W weight do not take the accessibility of the house into account when looking for a burglary target. Burglar 0, for example, has a weight of 0.0 and commits approximately the same number of burglaries in all houses, regardless of their accessibility. Burglar 10, on the other hand, has a weight value of 1.0 and commits many more burglaries in houses which are highly accessible. Results for Visibility_W (Table 18), Security_W (Table 19), TrafficVolume_W (Table 20), CollectiveEfficacy_W (Table 21) and Occupancy_W (Table 22) are similar. The results of Attractiveness_W (Table 23) are different to the other results as community attractiveness determines where an agent will start searching for a target but not actually where they will ultimately burgle.

The tests carried out so far have systematically shown that the PECS variables are controlling the agents' behaviour correctly and the burglary parameters are suitably influencing where the agents decide to burgle in the absence of a spatial environment. Changing parameter values does not result in any unexpected effects and model behaviour is not overly (or underly) sensitive to parameter changes. The next stage in model verification is to add a simple grid-based environment to the model and examine how it behaves.

Table 17: The number of burglaries committed by different agents in different houses by changing the Accessibility_W agent behaviour parameter (how important house accessibility is to the agent).

Burglar Accessibility_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	93	126	154	178	181	206	246	240	264	297	307
0.1	111	106	126	151	163	184	187	222	206	226	234
0.2	102	129	122	139	119	163	152	145	169	171	178
0.3	114	112	108	125	146	128	135	121	148	127	132
0.4	96	107	113	88	103	117	107	120	113	103	96
0.5	102	104	91	100	95	78	81	67	76	61	62
0.6	96	98	109	81	92	63	70	66	46	46	40
0.7	90	83	77	77	56	53	45	46	34	38	24
0.8	87	85	81	66	56	59	32	28	27	19	13
0.9	112	66	60	51	49	25	20	30	10	5	9
1	97	84	59	44	40	24	25	15	7	7	5

Table 18: The number of burglaries committed by different agents in different houses by changing the Visibility_W agent behaviour parameter (how much of a deterrent house visibility is to the agent).

Burglar Visibility_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	77	118	151	169	185	211	214	266	273	290	285
0.1	100	141	130	161	160	167	192	194	209	231	207
0.2	120	111	133	124	136	163	166	164	172	180	208
0.3	115	92	125	104	135	128	126	139	134	120	113
0.4	98	102	115	104	113	101	109	95	90	88	102
0.5	107	110	88	94	98	81	74	80	75	73	68
0.6	91	105	88	85	71	92	64	51	64	47	54
0.7	101	90	88	88	74	75	62	43	32	26	25
0.8	106	92	77	64	41	25	38	31	24	14	19
0.9	86	64	64	50	46	25	33	21	16	20	12
1	99	75	41	57	41	32	22	16	11	11	7

Table 19: The number of burglaries committed by different agents in different houses by changing the Security_W agent behaviour parameter (how much of a deterrent security is to the agents).

Burglar Security_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	95	131	136	162	189	207	239	244	269	273	287
0.1	100	120	164	182	151	188	186	215	198	239	249
0.2	128	122	117	133	152	156	134	134	186	171	167
0.3	107	126	118	107	120	122	141	144	147	133	134
0.4	94	102	105	88	119	101	93	113	98	90	96
0.5	86	82	101	100	71	88	93	74	71	74	60
0.6	92	116	75	86	82	53	66	61	49	46	48
0.7	106	67	80	83	78	54	51	56	39	35	32
0.8	95	83	68	54	52	51	31	27	17	22	17
0.9	104	79	58	53	54	53	38	18	14	9	7
1	93	72	78	52	32	27	28	14	12	8	3

Table 20: The number of burglaries committed by different agents in different houses by changing the TrafficVolume_W agent behaviour parameter (how much of a deterrent traffic volume is to an agent).

Burglar TrafficVolume_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	123	131	145	182	181	211	234	249	275	271	292
0.1	101	111	136	140	169	172	212	211	234	245	234
0.2	110	106	131	118	141	143	141	156	152	162	189
0.3	87	95	127	112	137	136	127	145	123	123	132
0.4	101	107	107	108	117	103	99	93	114	102	82
0.5	102	97	102	91	68	87	86	82	77	70	49
0.6	102	90	71	100	71	64	57	56	41	46	61
0.7	94	97	76	78	71	67	54	43	38	43	33
0.8	92	81	76	64	56	46	37	29	21	13	13
0.9	102	91	70	54	53	35	35	17	15	15	10
1	86	94	59	53	36	36	18	19	10	10	5

Table 21: The number of burglaries committed by different agents in different houses by changing the CollectiveEfficacy_W agent behaviour parameter (how much of a deterrent community cohesion is to an agent).

Burglar CollectiveEfficacy_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	98	143	134	176	199	227	227	253	260	282	305
0.1	94	110	129	166	159	179	192	224	210	222	253
0.2	91	119	129	125	152	151	151	142	176	171	159
0.3	90	118	119	121	138	103	126	122	124	114	118
0.4	105	114	98	103	103	113	108	96	81	94	92
0.5	100	90	112	87	88	75	87	81	86	89	71
0.6	110	85	80	72	66	79	73	71	62	45	53
0.7	107	84	80	83	69	47	53	26	39	33	24
0.8	88	82	82	61	46	50	38	39	34	24	12
0.9	114	78	68	55	39	45	20	31	19	14	7
1	103	77	69	51	41	31	25	15	9	12	6

Table 22: The number of burglaries committed by different agents in different houses by changing the Occupancy_W agent behaviour parameter (how much of a deterrent occupancy is to the agent).

Burglar Occupancy_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	107	132	155	174	186	200	214	233	283	288	283
0.1	86	125	146	125	175	193	189	204	199	222	239
0.2	88	101	127	163	166	151	176	162	180	158	172
0.3	109	117	119	117	117	132	115	141	100	132	137
0.4	82	110	77	124	84	112	114	98	100	92	98
0.5	85	120	91	97	101	65	87	90	75	79	67
0.6	101	88	82	98	77	70	75	66	59	55	36
0.7	122	71	96	69	62	53	41	41	46	33	33
0.8	100	78	83	48	54	61	32	36	21	28	16
0.9	118	96	68	43	44	35	32	21	27	13	18
1	102	62	56	42	34	28	25	8	10	0	1

Table 23: The number of burglaries committed by different agents in different houses by changing the Attractiveness_W agent behaviour parameter (how important community attractiveness is for an agent to start their search there).

Burglar Attractiveness_W	B0 0.0	B1 0.1	B2 0.2	B3 0.3	B4 0.4	B5 0.5	B6 0.6	B7 0.7	B8 0.8	B9 0.9	B10 1.0
0	107	98	109	95	113	102	98	110	107	94	94
0.1	89	87	101	89	90	111	100	81	111	98	91
0.2	96	100	88	107	99	91	101	104	94	102	126
0.3	97	92	96	95	90	105	105	121	113	104	95
0.4	92	121	105	82	107	96	101	77	107	85	86
0.5	111	84	100	123	109	90	114	89	99	101	97
0.6	103	103	99	95	98	91	102	99	110	92	103
0.7	103	112	99	105	103	108	106	110	85	90	108
0.8	118	91	106	121	88	107	77	99	88	108	104
0.9	89	100	95	86	97	86	105	98	94	99	95
1	95	112	102	102	106	113	91	112	92	127	101

3.2 Tests Using The Grid Environment

The previous section demonstrated that the model performs as expected in the absence of a spatial environment. The next stage in the verification of the model is to experiment with the addition of space: how will the agents behave when they actually have to travel around? The following tests will be performed:

1. Exploration of the spatial distributions of burglary under default conditions.
2. Varying the parameters that affect where an agent chooses to start their search: the Distance_W (how far to travel) and PreviousSuccess_W (how much more attractive areas are that the agent has already successfully burgled in) agent parameters.

Figure 21 depicts the layout of the environment. It has been kept simple to isolate the features being tested from spatial complexity (with the exception of the road network which will not be a simple regular grid in order to ensure the routing algorithms are working correctly, i.e. some roads are dead ends). It will consist of a single social location, drug dealer and workplace and contain one agent who always lives at the same home address. These locations are in opposite corners so that the agent must travel between them and build up an awareness of the environment (their "cognitive map") as stipulated by routine activities theory and crime pattern theory (Cohen and Felson, 1979; Brantingham and Brantingham, 1981). The different communities can be assigned different parameter values but are otherwise indistinct.

The first test is a general test of the addition of space to the model: all parameters have default values and all communities are identical. A single agent lives in the north-west corner of the environment and must burgle to satisfy their needs. As with previous tests the simulation was run for 30 simulated days and was run 50 times. Figure 22 illustrates the distribution of burglaries committed across all simulations. On the whole, burglaries are spread throughout the environment but are more dense around the agents' home.

Observing the distributions of burglaries, it is interesting that there do not appear to be clusters of burglary surrounding the social and drug dealer locations. This might be expected because these areas are very likely to feature in the agents' awareness spaces. Figure 23 illustrates the evolution of a burglar's awareness space in a

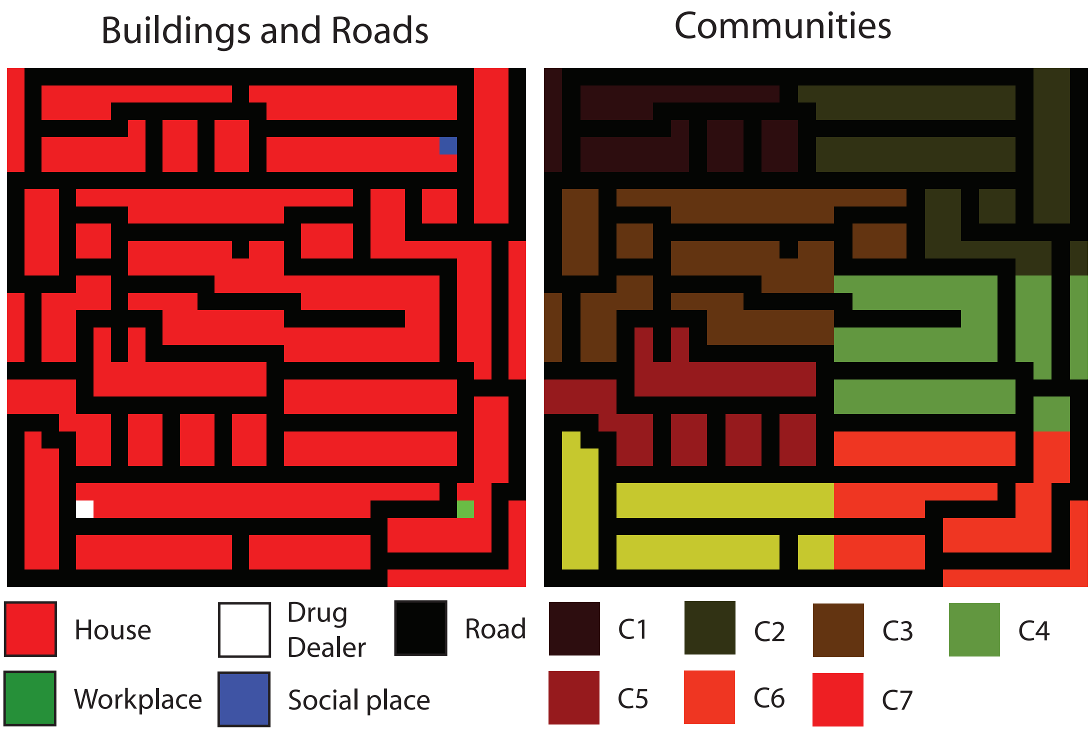

Figure 21: The layout of roads, buildings and communities used in the Grid environment sensitivity tests. The different types of communities can be assigned different parameter values depending on the experiment, although there are no inherent differences between them.

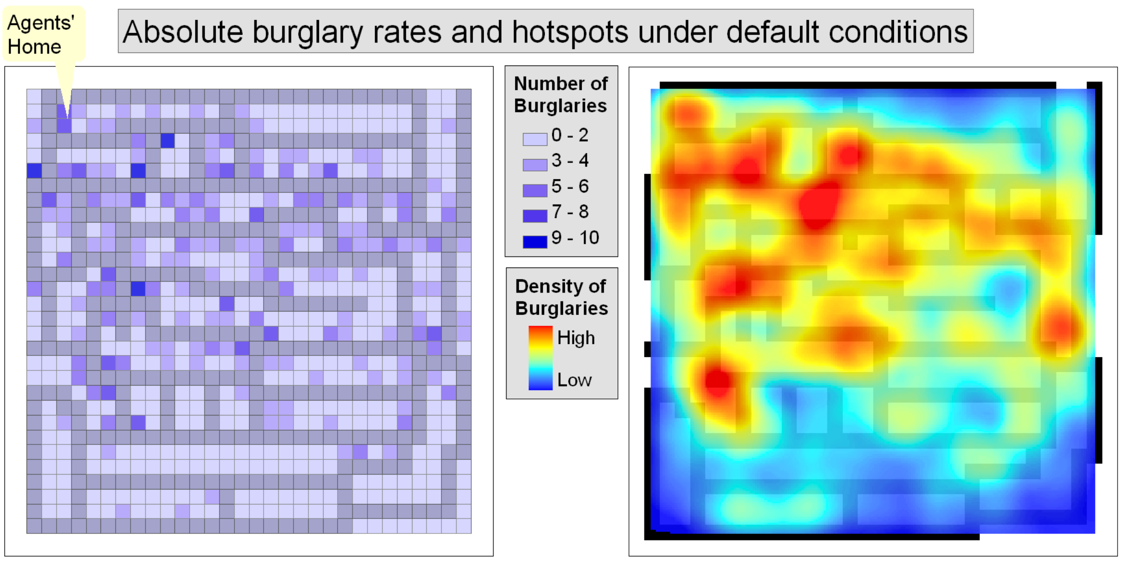

Figure 22: The distribution of burglaries with the default configuration (50 model runs). The absolute burglary count (left) and interpolated density using KDE (right).

single example model run. It becomes clear that approximately a third of the way into the simulation (10,087 iterations) the agent has nearly explored the environment in its entirety, so the awareness space in this case actually makes little difference to the choice of where to begin searching for a burglary target. As Section 3.3 will show, this is not the case in larger environments that more closely represent real urban configurations such as the GIS environment. Nonetheless, Figure 23 is able to illustrate that the agent's cognitive map is working correctly.

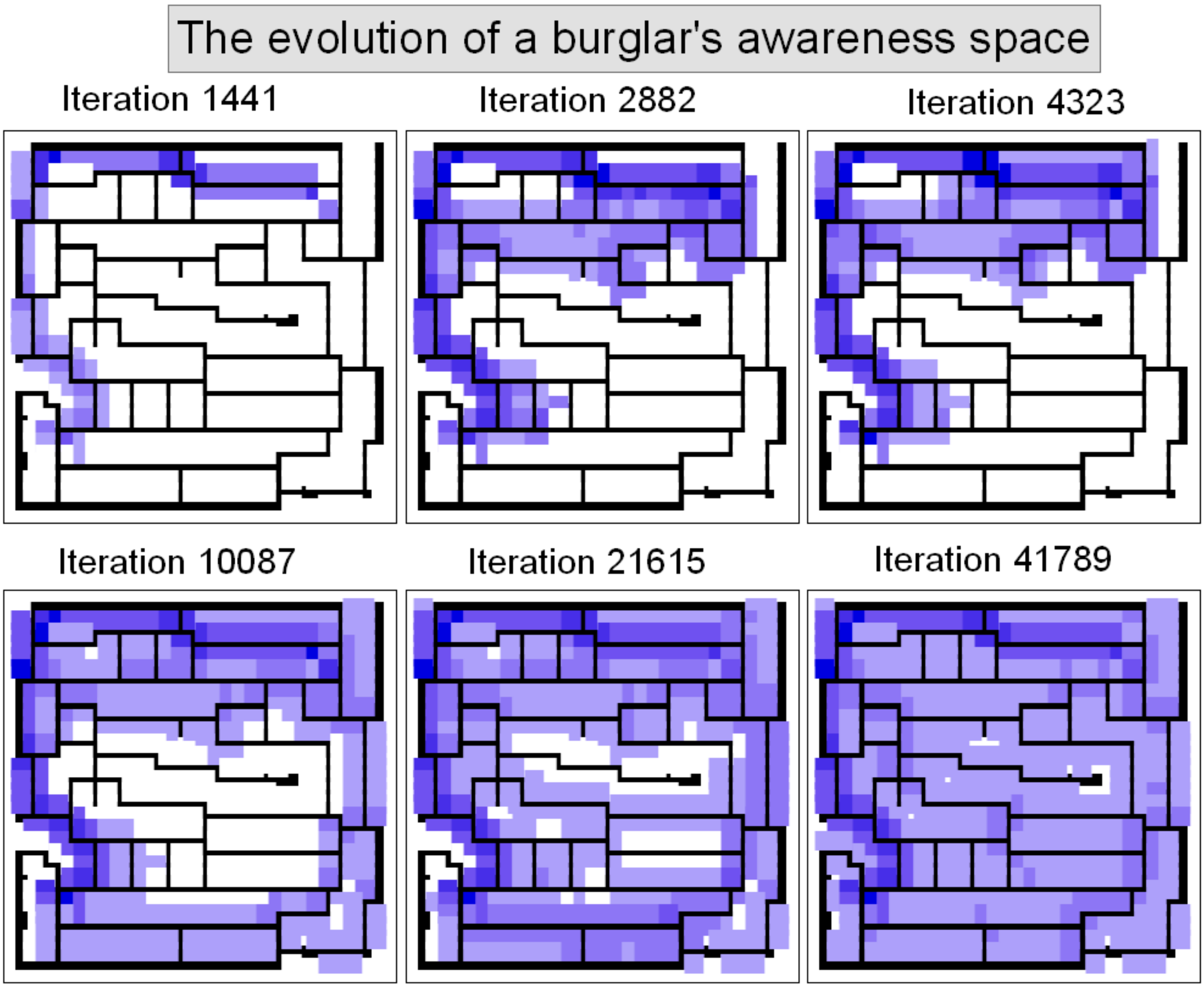

Figure 23: The evolution of a burglar's awareness space in a single example model run.

The reason that burglaries are more clustered around the agents' home, therefore, is because although all communities are perceived as identical the agent is still more likely to pass nearby properties at the start of the journey regardless of where they are actually going so the potential to burgle is always present. This is consistent with the literature, where the proximity to potential offenders is commonly associated with burglary risk (Snook, 2004; Bernasco and Nieuwbeerta, 2005).

The following experiment will alter this behaviour by changing the *Distance_W* parameter in order to determine where burglars are more likely to burgle when they are strongly/weakly influenced by the distance that a target community is away from their home. Figure 24 illustrates the results in the form of hotspots and absolute number of burglaries. However, it appears that altering the parameter has made little difference to the results: the patterns produced by the model when the parameter was altered between 0.0, 0.5 and 1.0 are similar. The reason for this is non-trivial. When deciding where to search for a burglary target, the agent assigns a value, l, to each community, a, based partly on its distance [3]:

$$l_a \propto Distance\_W \cdot \frac{1}{dist(c,a)} \tag{15}$$

where $dist(c,a)$ represents the distance between the agent's current location and the target community. It is not unsurprising, therefore, that changing the value from 0.5 to 1.0 has no effect; as there are no other parameters influencing the agents' decision, the attractiveness of the communities, relative to each other, will be the same. A value of 0.0 might be expected to alter the pattern, however, because that would discount the influence of distance completely and thus increase the spread of burglaries away from the agent's home. The fact that this does not happen is indicative of the size of the environment: the agent is able to search most of the environment without becoming desperate for burglary so where they actually start the search is of little importance.

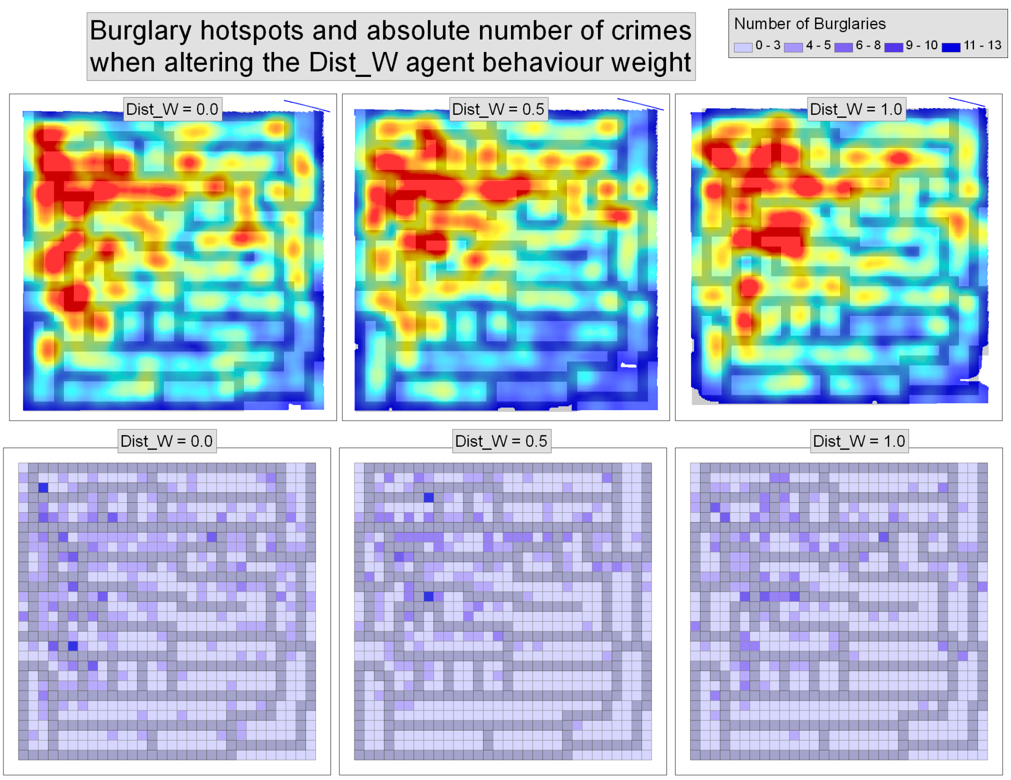

Figure 24: Results when altering the *Distance_W* agent parameter (total of 50 model runs): density using KDE (top) and absolute counts (bottom).

[3] Equation 15 in full is:

$$l_a = A \cdot \frac{1}{dist(c,a)} + B \cdot attract(h,a) + C \cdot socialDiff(h,a) + D \cdot prevSucc(a)$$

where $dist(c,a)$ represents the travel distance to the community a, $attract(h,a)$ is the relative attractiveness of a, $socialDiff(h,a)$ represents the social difference between their home community and a, $prevSucc(a)$ is the number of times they have successfully burgled in a and $A-D$ are internal weights that the burglar applies to each paramter.

To ensure that allowing the model to run for a longer period of time does not change the effect that altering the Distance_W parameter will have, Figure 25 illustrates the results after running the model for sixty days instead of only thirty. Again the results are similar regardless of the value of the Distance_W parameter.

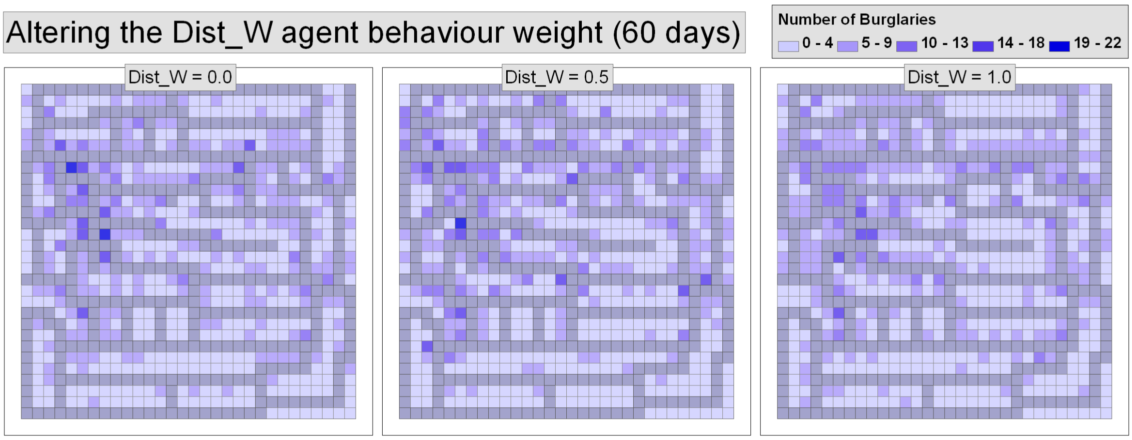

Figure 25: Results when altering the PreviousSuccess_W agent parameter using a longer simulation run-time of 60 days (total over 50 model runs).

The next parameter to be tested is the PreviousSuccess_W parameter, which determines how much an agent is drawn to communities which they have previously burgled in. With high values for this parameter more clustering is expected as agents return to the same areas repeatedly to look for targets. Figure 26 illustrates the results. Again, however, the size of the environment leads to initially unexpected results as the burglary distributions look similar. To ensure that this result is not an artefact of summing all burglaries over a number of model runs some results from individual model were also examined but these also showed no discernible patterns. Again this result is likely to be due to the size of the environment; the agents are able to investigate most houses in search of a target so it matters little where they actually *start* their search.

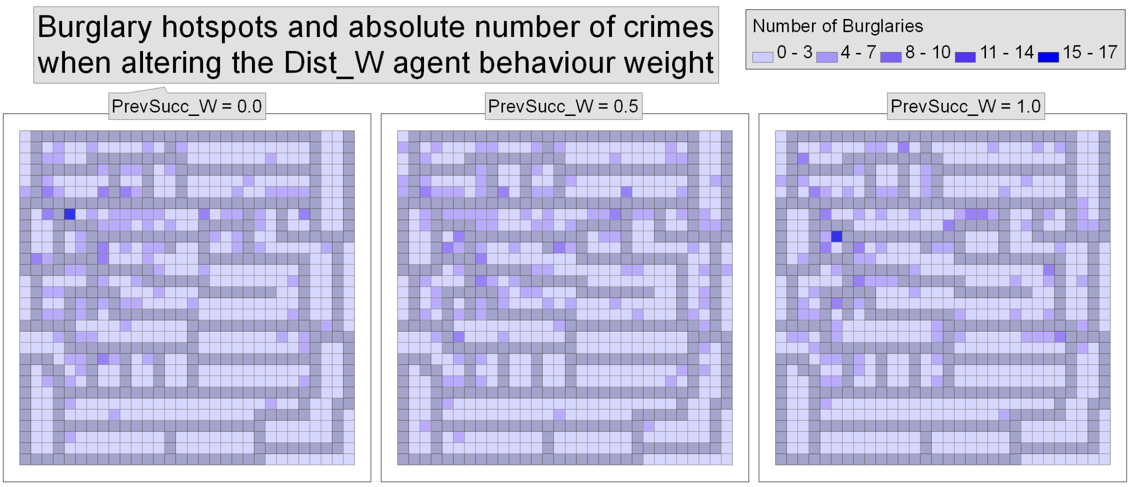

Figure 26: Results when altering the PreviousSuccess_W agent parameter (total over 50 model runs).

As the previous results are somewhat inconclusive it is not necessary to sensitivity test the Sociotype_W parameter. Although (as with other experiments) it will likely vary the location where the agents start their search there is nothing to suggest it will influence where the agents finally burgle.

Overall, the results show that the experiments demonstrate utility. Importantly they are able to show the model can work in an environment other than the null environment without any changes to the agents. Furthermore, they have demonstrated that the burglars' awareness space is working properly. In particular, notice that in the map of the agent's awareness space (Figure 23) there is a dead-end road in the centre of the environment which is the last area that the agent discovers. This demonstrates the advantages of an agent-based model and burglars with an awareness space: houses on quiet roads without through-traffic might be less of a burglary risk simply because potential burglars are unaware of them. Other types of models will find it extremely difficult to include these types of effects that are fundamental to modern criminology theory (Cohen and Felson, 1979; Brantingham and Brantingham, 1981).

Another advantage of the tests is that it has become clear that there is a risk that where the agents start their search is immaterial if they are able to search a very large area before deciding to burgle. This risk can be mitigated in future experiments and provides evidence for the benefits of increasing model complexity gradually.

3.3 Tests Using The GIS Environment

The previous sections have illustrated that in the absence of a Euclidean space the model performs as expected to. The addition of a simple Euclidean space illustrated that the model is able to work with different environment configurations without changing the behaviour of the agents or other environment objects (such as buildings, communities etc.). The following section will increase complexity further and introduce a space that closely represents a real city: a GIS environment.

It would be ideal to generate a large hypothetical city designed solely for the purposes of testing the model. However, this would be an extremely difficult task so instead the part of Leeds which will be used in subsequent crime prediction experiments will be used. This does not detract from the utility of the tests because it is the *relative* results which are important. For example it does not matter where a burglar is actually burgling, only if they move locations when a parameter is changed.

The most important aims of this section are:

- to establish that the model still works properly using the new GIS environment;
- to determine whether or not the assumptions made in the previous section regarding the effects of community parameters on burglary locations are indeed accurate (burglars would have been influenced by the community parameters but the grid environment was too small);
- to test an important feature of the model that is otherwise untested: transport routes.

To meet the aims listed above, three experiments are conducted. The first uses the default model configuration (described below) to determine how an offender behaves in the GIS environment. The second changes the *Distance_W* parameter so that the agent prefers to stay near their home rather than travelling to distant communities. In the final experiment the effects of the transport network are tested.

Figure 27 summarises the environment, showing the locations of different types of buildings and the transport route which will be used in the second experiment. All communities and buildings have default values for all their parameters. A single agent is created at the location specified. Initially they have no access to transport and must walk through the environment, although in the second experiment they are able to utilise the available public transport.

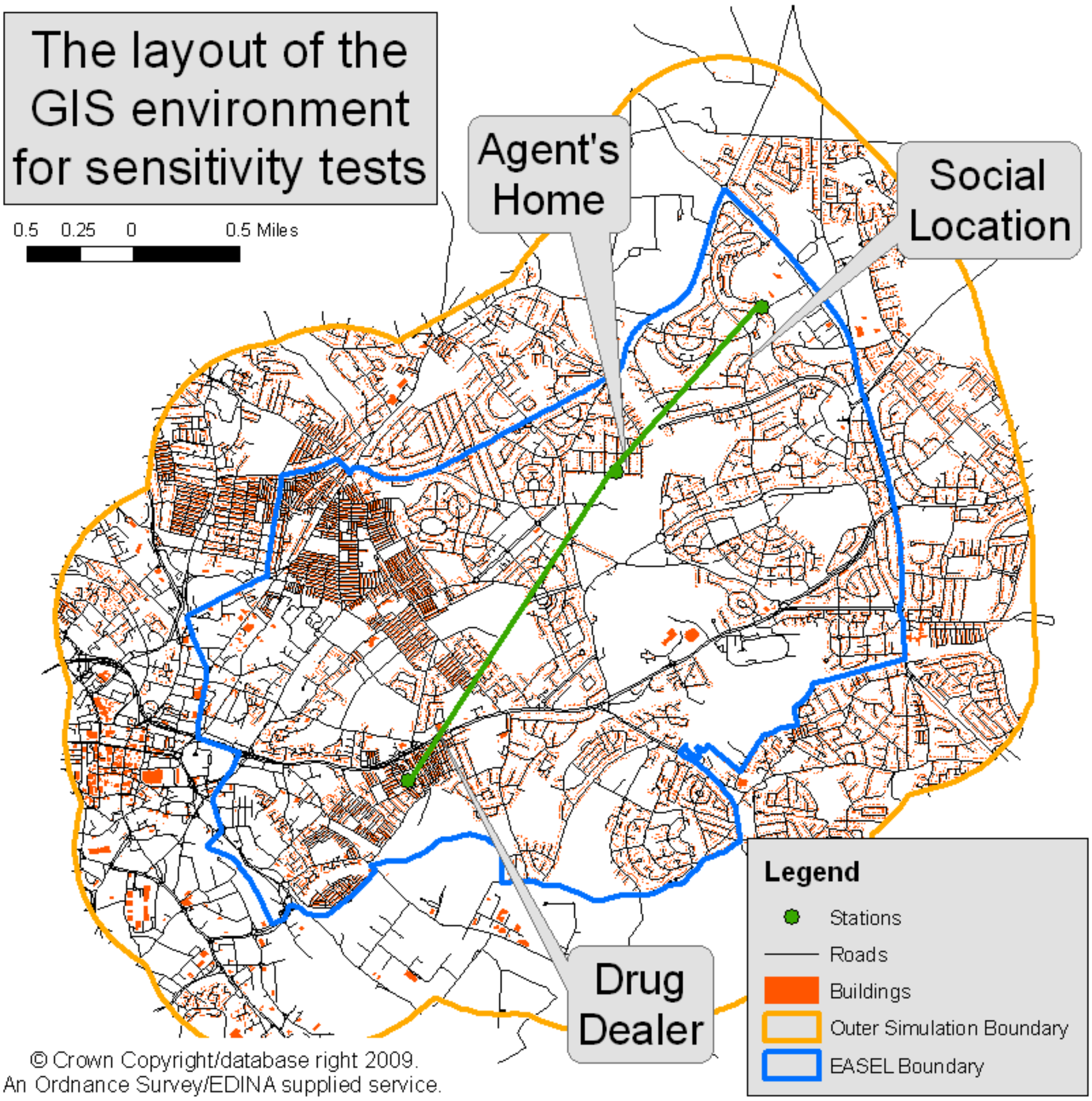

Figure 27: The layout of the environment used in GIS sensitivity tests.

Figure 28 illustrates the results of all experiments. All models were run 50 times; the figure shows the aggregated burglary counts and the hotspots that were produced using the kernel-density algorithm (cell size 10m and bandwidth 100m). The most striking result is that when the agent has access to public transport they use this to travel between their home, their drug dealer and the social location. When the agents are using transport they do not add the houses they pass to their awareness space which explains why the locations of burglaries are so much more heavily clustered. This seemingly simple result is significant because it illustrates the power and novelty of the model. No other published crime models are able to account so cleanly for advanced criminology theory such as Brantingham and Brantingham (1984)'s crime pattern theory. Using the model in this way will certainly be useful for the forthcoming crime forecasts. Figure 29 illustrates this finding in more detail.

Altering the *Distance_W* agent parameter also has the desired effect on the model, although it is not pronounced. With low values the agent is not concerned with how far they travel before starting their search. The reason that burglaries are not even more distributed is because the agent's awareness space does not extend far beyond the areas around their anchor points and the routes between them. When the value is increased, however, more burglaries are committed closer to the agent's anchor points. This is evidenced by an increased density around the agent's home in Figure 28 and Table 24 which displays the average distances that crimes were committed away from the agent's home or the nearest anchor point.

Table 24: The average distances that crime were committed away from the agent's home and all anchor points. Distances increase with increasing values of the *Distance_W* parameter.

Distance_W Value	Mean Distance (m)	
	From Home	From Nearest Anchor Point
0.0	1121	755
5.0	995	709

Verification – Conclusion

This section has systematically tested numerous model parameters. It was established in Section 3.1 at what point parameters could "break" the model by causing a PECS state variable value to drop to zero. Others were tested to determine whether or not they had the desired effect on the model and also how sensitive the model is to their value. Furthermore, complexity was added to the model systematically through the use of different types of environment. The experiments have shown that the model performs as expected given its definition and configuration and is not overly sensitive to any of the parameters. Confident that the model has been programmed correctly and is performing as expected, it is possible to begin calibrating the model.

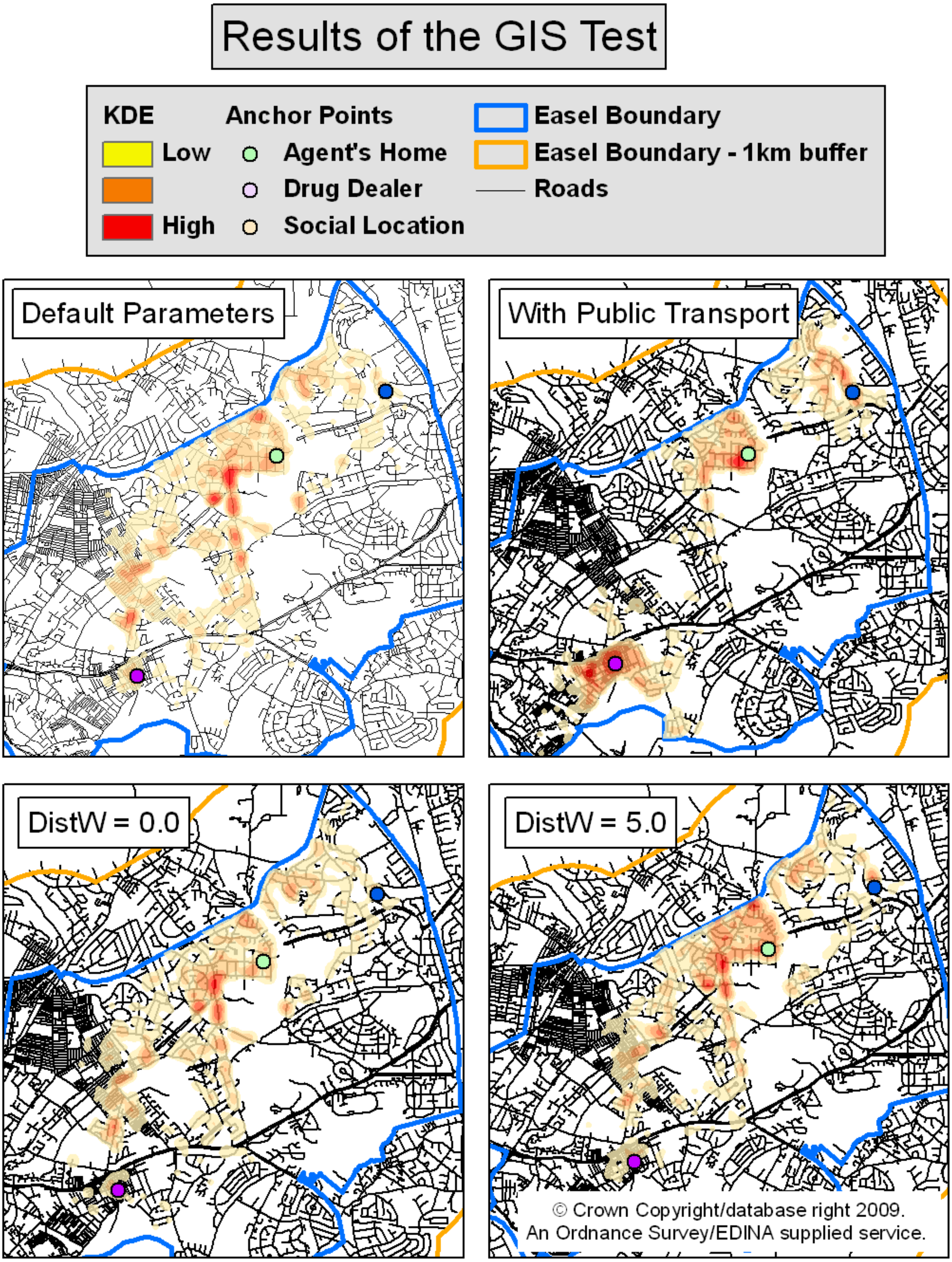

Figure 28: The GIS environment test results (total of 50 model runs): normal conditions (upper-left), with public transport (upper-right) and altering the *Distance_W* parameter (bottom).

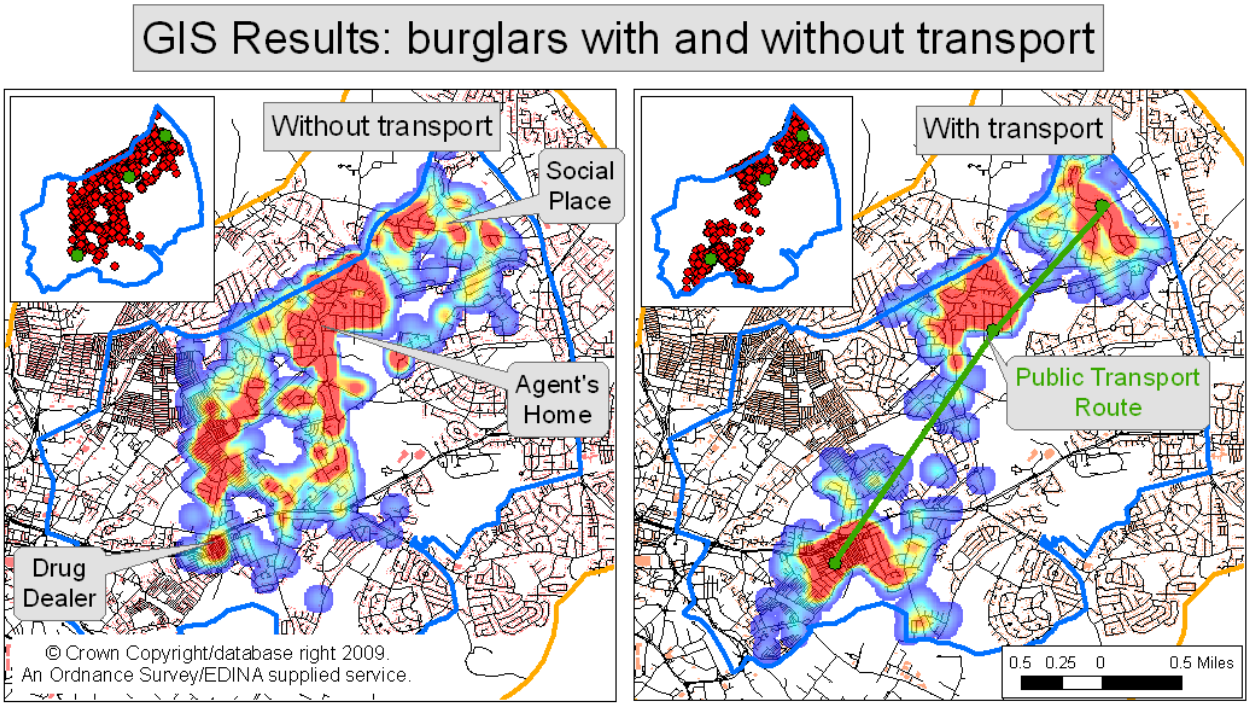

Figure 29: The different burglary patterns produced when a virtual burglar has access to public transport and when they must walk.

4 Calibration

Calibration is the process of configuring a model's parameters to match some field data (O'Sullivan, 2004). In many cases, a single value can be found which summarises how closely the simulated and field data match, this is commonly called "fitness" (Gilbert and Terna, 2000). Section 2.2.7 reviewed statistics that can be used to establish fitness in this application. Algorithms can then be used to search the model parameter space looking for the combination of parameters which produce the greatest fitness (i.e. the combination that produces outputs that are the closest match to field data). Examples of the use of a genetic algorithm to optimise an agent-based model exist in crime (Malleson et al., 2009) and retail (Heppenstall et al., 2006) among others. Unfortunately, using an automatic calibration routine is beyond the scope of this research for two reasons. Firstly some parameters that will have a strong influence over model results do not translate to a numerical scale (such as the starting locations of the burglars or the addresses of drug dealers). More importantly, however, is that a conservative estimate for the run time of a single automated calibration routine running on a high-performance computing grid is more than thirty days[4]; much longer than is realistically acceptable.

Instead, the model will be calibrated manually, finding the best fit to known data by visually and mathematically comparing results and adjusting parameters accordingly. Here, model fitness will be determined by producing graphs of the SRMSE and R^2 errors at different spatial resolutions as discussed in Section 2.3.

4.1 Calibration ("Base") Scenarios

The following scenarios are termed "base" because they represent the foundation on which later forecasting experiments will be based. They are built on criminology theory and real data analysis and will simulate the situation in 2001 (the time of the most recent UK census). All environment variables will have their values calculated from the census and other data. Figure 30 summarises where the different types of buildings are located (generating drug dealer locations is discussed shortly). The simulation will cover the south-east Leeds area plus a 1km buffer to limit boundary effects such that all crimes committed in the buffer area will be disregarded. Ideally the buffer would be larger such as 3km which is more than the average distance travelled by burglars found in many studies (Baldwin and Bottoms, 1976; Bottoms et al., 1992; Wiles and Costello, 2000; van Nes, 2006; Bennell et al., 2007). However, computing requirements increase exponentially with the radius of the environment (due to the the increasing number of offenders and environmental objects) and 1km is an acceptable compromise. The buffer area is considered the same as the inner area by agents, but when results are collected only crimes committed within the inner area will be counted to avoid boundary effects.

The first task is to determine how many burglar agents should be created and where they should live. The best estimate for offender home locations can be found in the offender crime data which pinpoints where convicted offenders live and where they committed the crime(s) they were convicted of. Only agents who live in the simulation area *and* commit a crime there are included in the simulation.

Once the relevant offenders have been found, points are aggregated the up to the output area level to determine, approximately, how many offenders each output area (OA) will house. Then when the model is initialised this many offenders are created in random houses in the OA. This operation is performed to limit the dangers of

[4]Assuming that the automated routine requires 100 iterations to reach optimal fitness (Malleson, 2006) and each model requires 50 separate runs to ensure the results are stable (discussed in Section 3.1) that is $100*50 = 5000$ individual runs. If a node on the high-performance computing grid used by this research requires 20 hours to complete a model run and there are a total of 128 nodes available the total time is $(5000*20)/128 = 781$ hours, or 32.6 days. Furthermore, this is a conservative estimate because the compute resource will be required by other users at the same time, reducing the number of nodes available to this researcher.

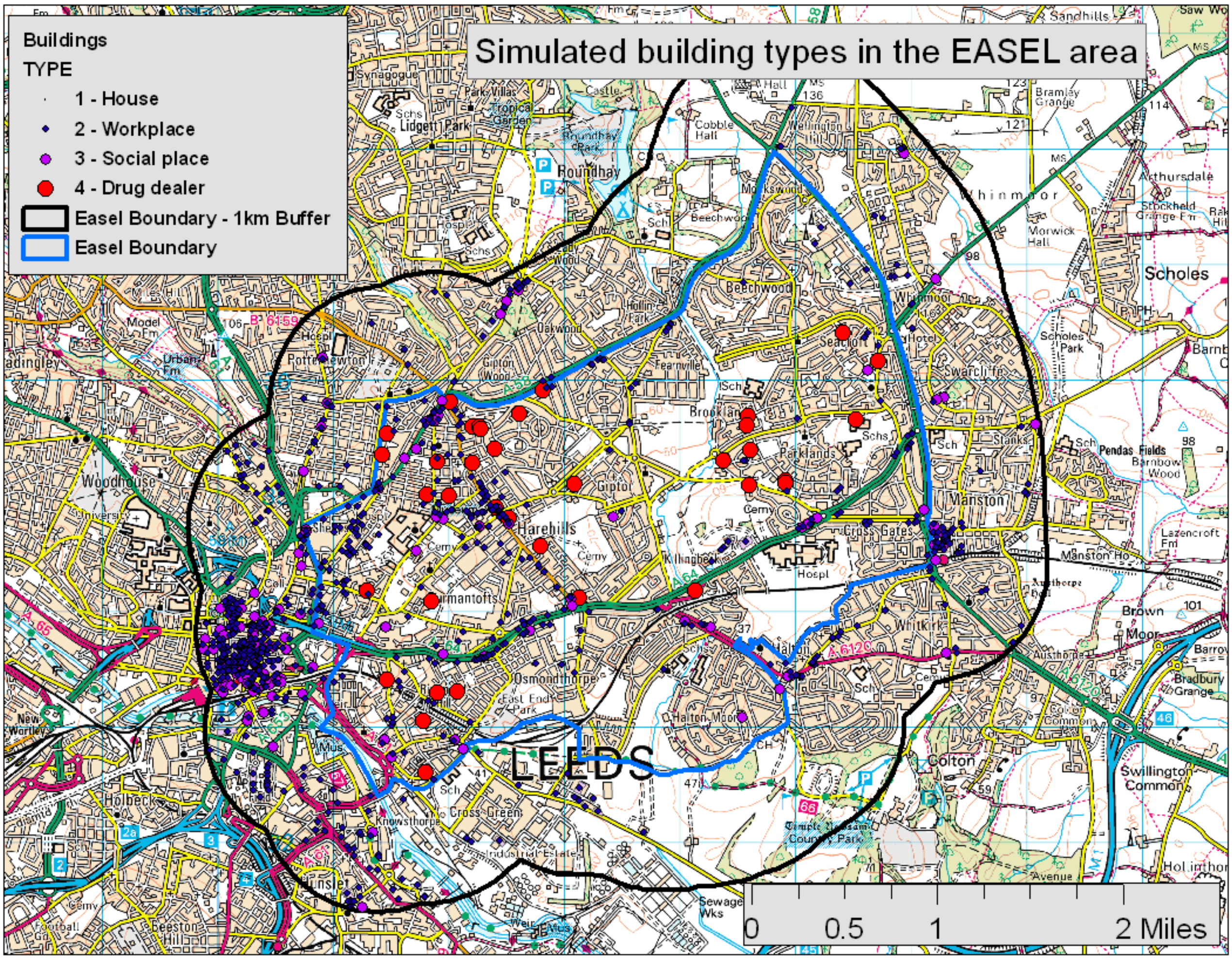

Figure 30: The locations of different building types, established from census and land-use data.

holding sensitive individual-level data (particularly because the model will be executed on external computer systems). Having established an estimate for how many offender agents should live in each area, it is also necessary to estimate where the drug dealer locations are. This can also be estimated from the crime data.

The final element of the scenario relates to how an offender chooses where to travel if they need to visit a drug dealer or to socialise. This is one of the most difficult features to estimate as there is very limited data which can assist with the assumptions. With regards to drug dealers, it is decided that the agent will chose any drug dealer that they know about at random and travel there. It is likely that in reality a person builds a preference for a certain dealers or addresses and visits these regularly, but this avenue of exploration is not in the scope of this work. With regards to social locations, it is assumed that an agent is more likely to travel to a social location that is in a community of a similar type to their own. Again this is likely to be too simplistic but can be investigated further if required.

Analysis Framework

The results of each simulation will be analysed using the same processes. Previous experiments ran the model 50 times so this practice will be continued here. Each scenario will be analysed systematically in the following manner:

1. Determine whether or not the results across all 50 models are consistent. This can be accomplished by producing a graph of the L function for all models; if the L functions are similar then the point distribution is similar. Hotspot maps for some randomly chosen models can be provided to verify this.
2. Determine whether or not the simulation has reached equilibrium. As stage 1 established that all models produced similar results, only one randomly chosen model needs to be examined as there is nothing to suggest that others will differ. Equilibrium can be determined visually by examining the point density at different points in the simulation and mathematically by graphing the difference between L functions produced at different time periods. I.e. if there is a difference between the L functions produced at time t_1 and t_2 then the simulation has not reached equilibrium by time t_2.
3. Finally the results can be compared to the expected data. This will be accomplished using the expanding cell algorithm; producing a graph of SRMSE and R^2 errors at different resolutions and mapping the differences.

4.2 Base 1 – Default Conditions

This section will present the results of the first scenario. Figure 31 maps the simulated burglary rates for four randomly chosen models and Figure 32 graphs the spatial distribution of all models in the form of L functions. Both figures suggest that all models are producing similar results and it is safe to continue analysing the model results, assuming that all models are similar.

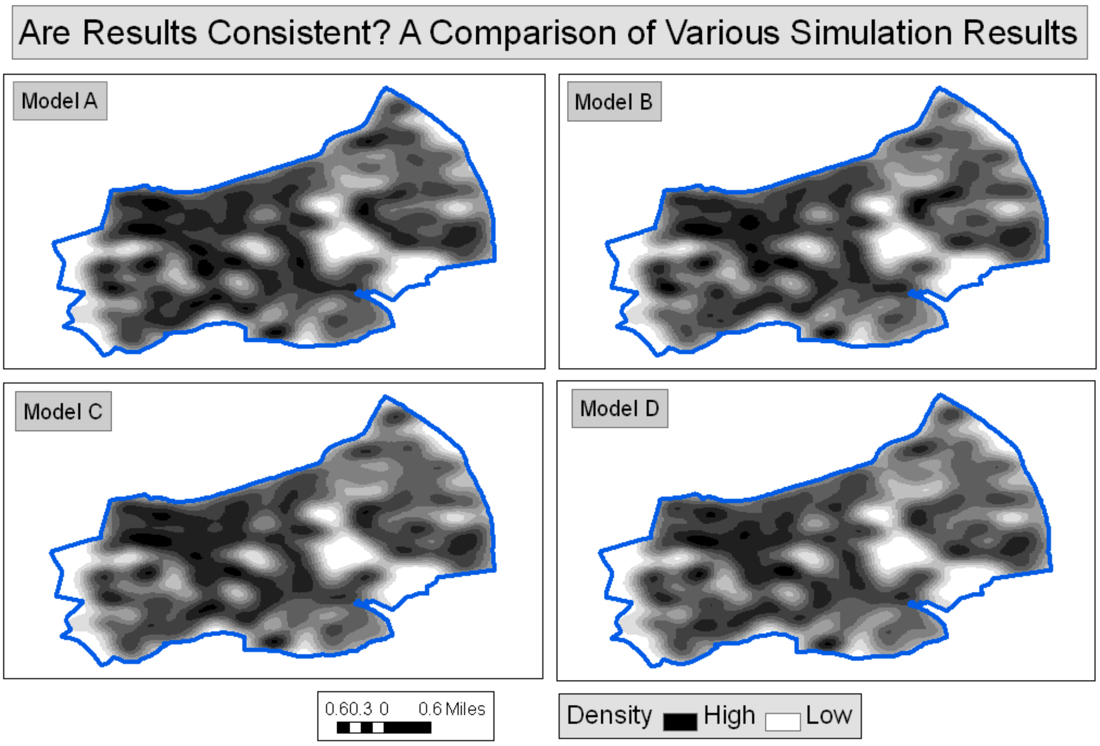

Figure 31: Map comparing different Base 1 models.

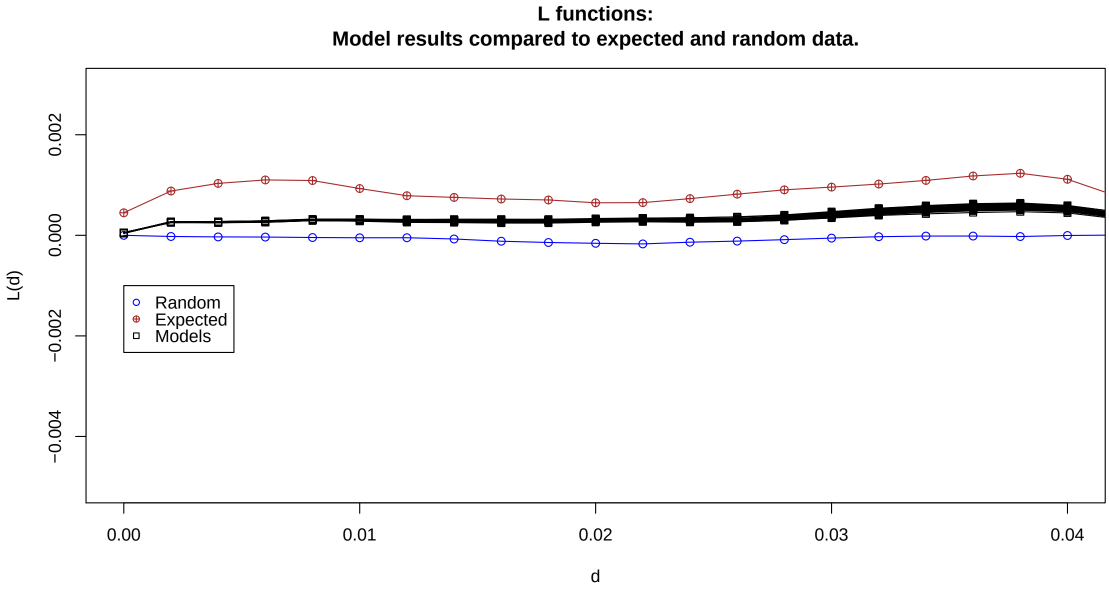

Figure 32: L Functions for a number of Base 1 models compared to expected data.

To determine whether or not the simulation reached equilibrium, Figure 33 maps the density of burglaries at four different points in an example simulation. Furthermore, Figure 34 graphs the difference in the L functions for the different simulation times. Both figures suggest that the simulation has reached equilibrium: the maps because they look similar and the graphs because the difference between the third and last time periods is very small.

Figure 33: Map comparing hotspots produced by a single Base 1 model at different simulated times.

Having determined that the simulation results are consistent and have reached equilibrium, Figure 35 compares the burglary density of a single model to that of the expected data. In both (and all subsequent) cases, the same parameters were used in the kernel-density algorithm (a 350m bandwidth with a cell size of 20m – chosen through trial-and-error because they produced the most appropriate visual summary). To supplement this, the expected and simulated point patterns are compared directly using the expanding cell algorithm. Figure 36 maps the results of the analysis at two different resolutions and Table 25 illustrates the global errors found by the expanding cell algorithm at the two mapped resolutions.

Table 25: Base 1 model errors at different cellular resolutions

Number of Cells	Resolution (Cell Area, km^2)	SRMSE	R2
121	0.65	0.84	0.61
441	0.18	1.05	0.5

Observing Figure 35, the model does not seem to be simulating burglary accurately (although this is to

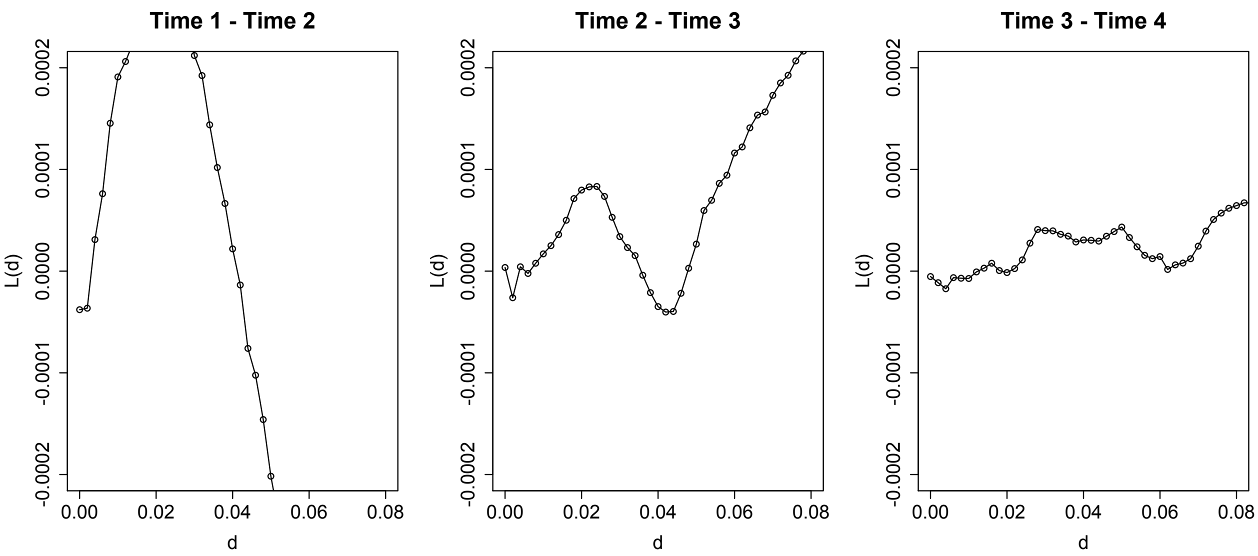

Figure 34: Graph of the difference in L functions at different points in a Base 1 simulation.

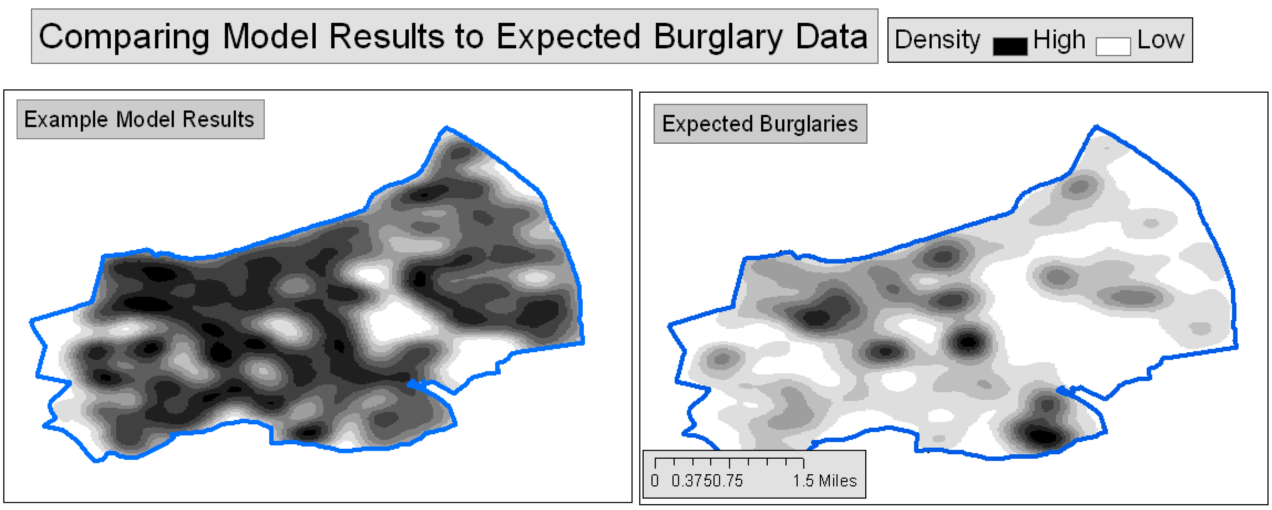

Figure 35: Map comparing Base 1 model results hotspots to expected data.

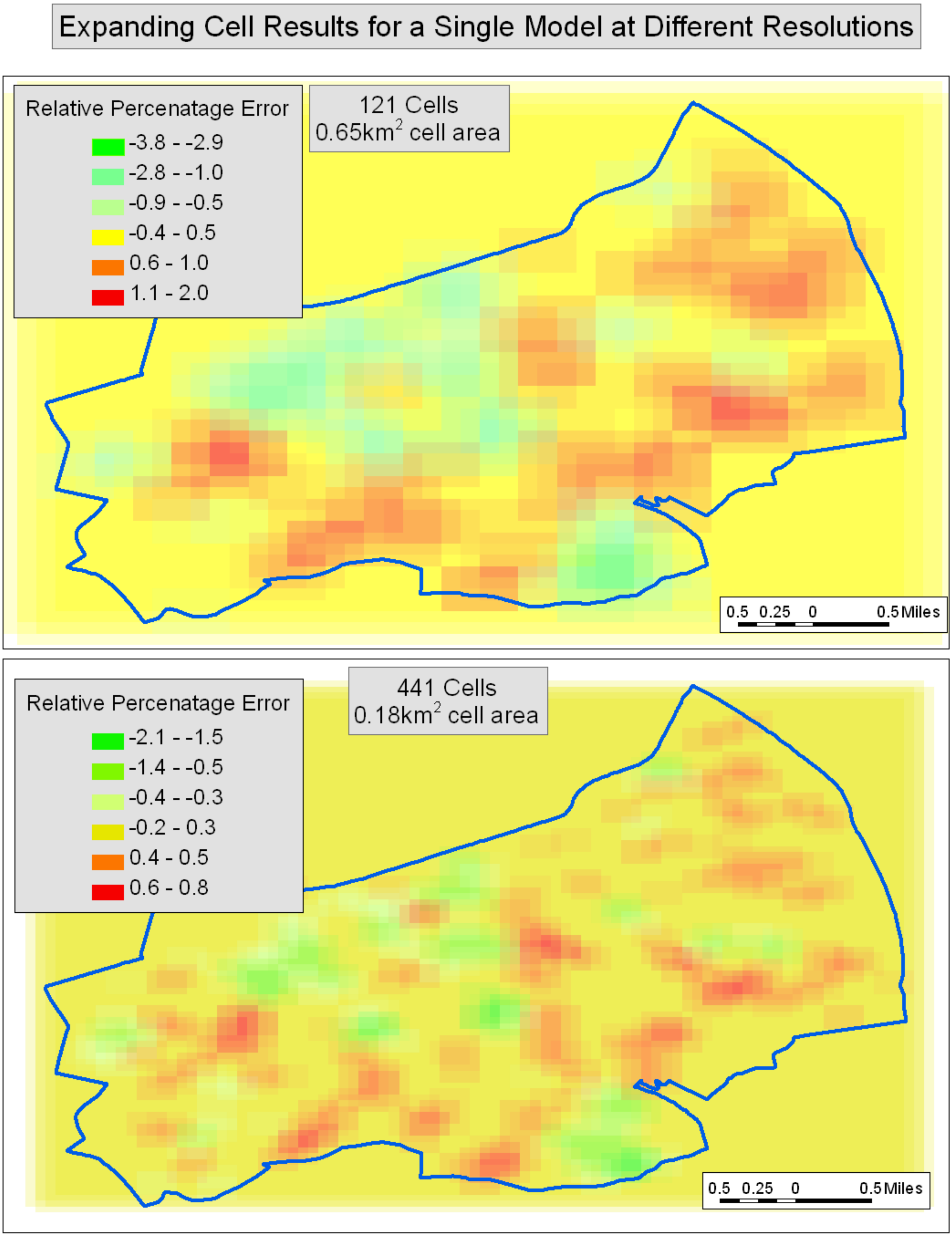

Figure 36: Expanding cell maps for a Base 1 Scenario

be expected with an un-calibrated model). Simulated burglaries are distributed throughout the environment whereas in the expected data they are more concentrated in hotspots. In observing the expanding cell map (Figure 36) another discrepancy becomes apparent: there are a small number of cells in which the model is significantly under-predicting the number of crimes. This is further illustrated by the frequency distribution of cell errors in Figure 37. The large model under-estimations relate mainly to a single neighbourhood in the south-east of the EASEL simulation area called 'Halton Moor'.

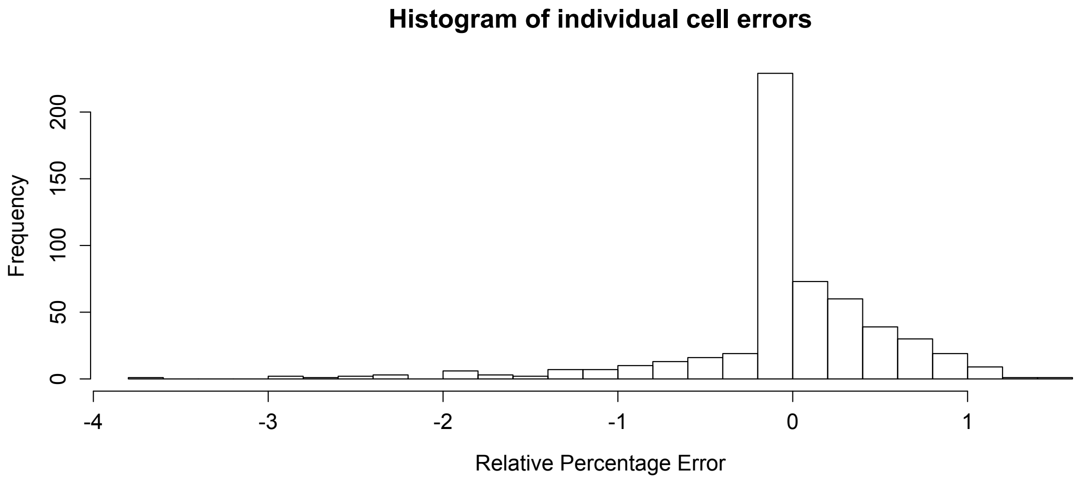

Figure 37: A histogram of all expanding cell Base 1 errors at the 0.85km^2 resolution.

Before continuing with the calibration process, it is important to determine why the errors are so large in this particular area. It is possible that the crime data are actually misrepresenting the number of crimes committed in Halton Moor. For example, a geocoding error might have placed a number of crimes in the area that do not belong there. A manual inspection of the data suggest this is not the case and, as Figure 38 illustrates, the crimes appear to be realistically distributed (there is not a single address that accounts for all crimes in the area for example). Therefore it appears that the hotspot in Halton Moor is a realistic portrayal of crime in the area and not a result of data errors.

It is also possible that a large number of offenders are travelling to the area from a long distance away. This has been found to occur if an area is particularly attractive to burglars. If this is the case, and the offenders in the real world are travelling from outside the area that is being simulated, then they will be accounted for in the simulation. This would explain why the model was not correctly simulating crime in the area. However, Figure 39 visualises the offender dataset and demonstrates that most of the crimes are committed by offenders who live within the area of the hotspot. Although the offender data are extremely sparse and probably a poor indicator of offending activity, it nevertheless gives no indication that crimes are being committed by people from outside the EASEL area.

As the crime data fails to provide a reason for why the hotspot should exist in Halton Moor, it is possible to search for one in the community and census data instead. The Output Area Classification (Vickers and Rees, 2006) classifies communities into distinct groups based on the 2001 UK census. Examining these groups in EASEL and Halton Moor (Figure 40) does not reveal any reasons for the hotspot. At the supergroup level (the highest level in the classification hierarchy) there are no particular differences between Halton Moor and

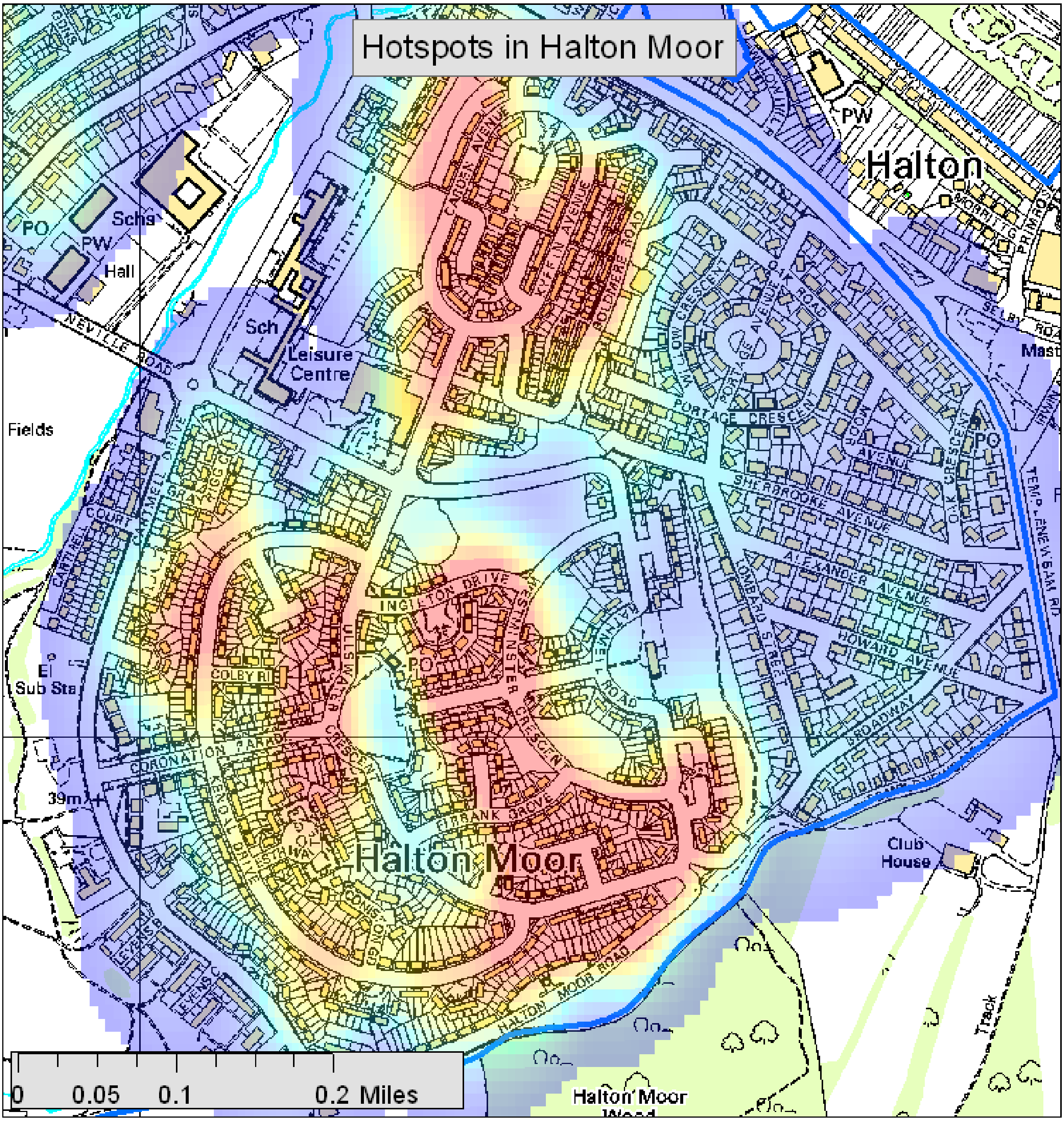

Figure 38: Real crime hotspots occurring in the Halton Moor area.

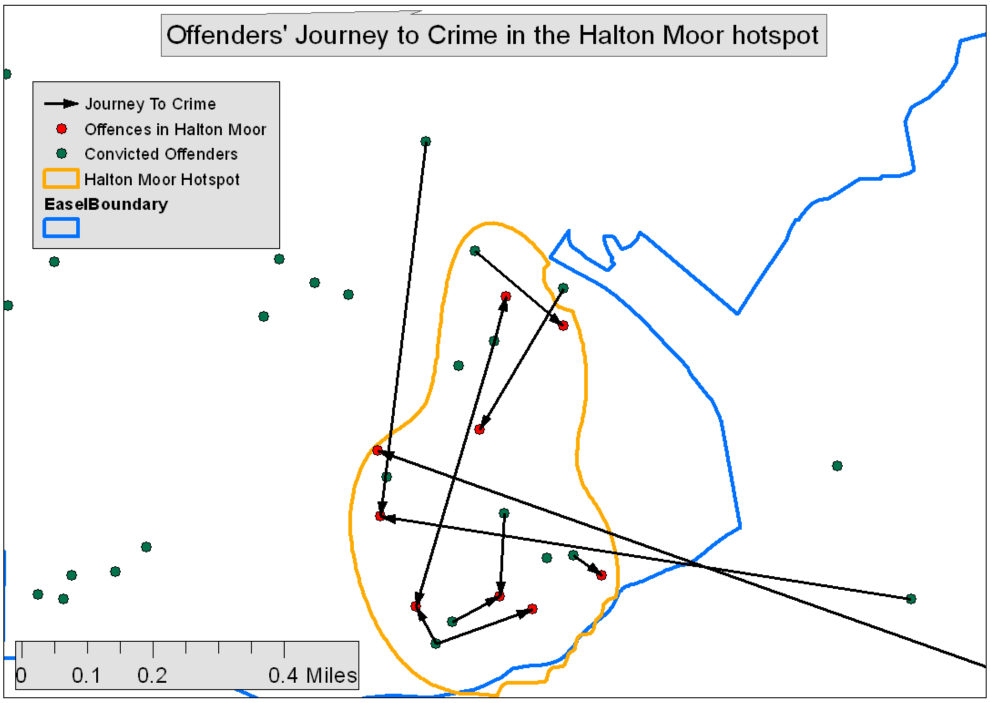

Figure 39: Journey-to-crime in the Halton Moor area. The starting locations of nominals who were associated with a crime that occurred in the Halton Moor hotspot, from the offender dataset.

other parts of EASEL. For example, most of Halton Moor is made up of "blue collar" and "constrained by circumstances" groups but these are prevalent throughout the EASEL area.

In summary, the available data provides no explanation for why the model is unable to predict the existence of a crime hotspot in the Halton Moor area. To try to gather further information, experts at Safer Leeds were consulted regarding the hotspot. It was suggested that, at the time, the area had some severe social problems and burglary was thought to be used often as a form of intimidation. However, the usual assumption (as implemented in the model) is that burglars are largely motivated by monetary gains. In this case, the model is able to demonstrate where, specifically, common assumptions about burglary fail. This is an extremely interesting result. In the meantime, the process of calibrating the model will proceed, disregarding the inability to simulate the Halton Moor hotspot.

4.3 Base 2 – Faster Security Decreases

Section 4.2 found that the model was unable simulate a particular crime hotspot and that this was possibly a result of incorrect assumptions regarding burglar motivation. Although the model has been designed to incorporate different types of motivation, there are other means of calibrating the model more generally first. Therefore the absence of a hotspot in Halton Moor will be disregarded for the time being and alternative methods will be used to improve the accuracy of the model. The first of these, which will be tested in this scenario, relates to the dispersion of burglaries.

Figure 32 provided a graph of the L functions for all the Base 1 simulations and compared these to the L function produced by the expected data. As the L values for the simulated data are less than the expected data at all distances, there is less clustering in the model than suggested by the expected data. This is also apparent by

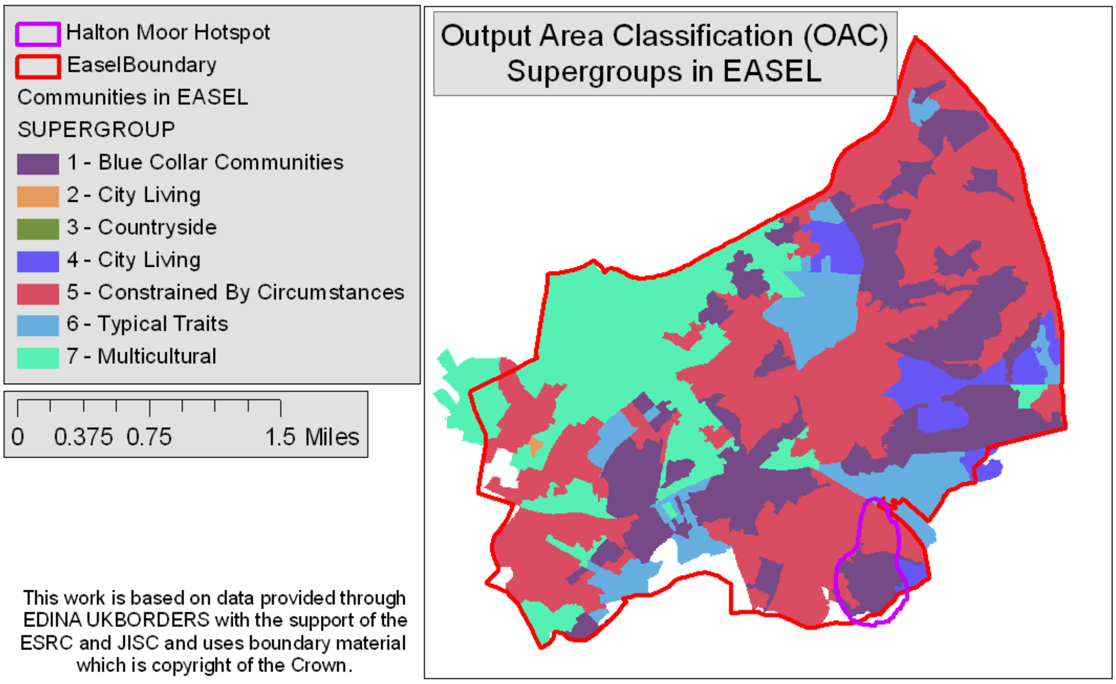

Figure 40: The Output Area Classification (OAC) groups in the EASEL area. Illustrating supergroups (the largest group in the hierarchy).

visually comparing the density maps (see Figure 35). Furthermore, Figure 41 demonstrates that in the model, there are very few buildings with more than one burglary but in the real data some buildings are repeatedly victimised. Therefore, burglary in the model is less clustered than expected.

One reason for the greater dispersion might be due to the effects of household security. When a property is burgled, it has been assumed that both the burgled house and its neighbours increase their security levels in response to the new threat. Even though agents are subsequently more strongly attracted to the area (because they remember that they have been successful there), the high security levels make it more difficult to actually commit a burglary in the area. A closer inspection of security changes indicates that, in this manner, security in the simulation increases indefinitely. Figure 42 demonstrates this with a frequency distribution of the security levels in all buildings at different time points in the simulation. Initially most buildings have very low levels of security, but over time security increases indefinitely.

By making security decrease more rapidly, it should be possible to address this error *and* reduce the dispersion (as there will no longer be areas with extremely high security that are effectively impossible to burgle). This can be accomplished by changing two parameters. Firstly the rate at which security decreases is adjusted so that it halves over the course of a week (reducing by $(\frac{1}{2})^{\frac{1}{7}}$ per day) rather than simply by decreasing by 0.5. Secondly, the amount that security will increase in surrounding properties following a burglary is reduced. Now, instead of an increase of 2 units, houses within 1 distance unit of a burglary will only receive an increase of 0.5 units (and this value decays as normal. These numbers were established through trial and error as empirical evidence to support their value is limited. For example, quantifying how quickly a household becomes complacent following a burglary, if they do at all, is non trivial and beyond the scope of this research. However this is the advantage of these types of models; it is possible to experiment with these types of parameters in the

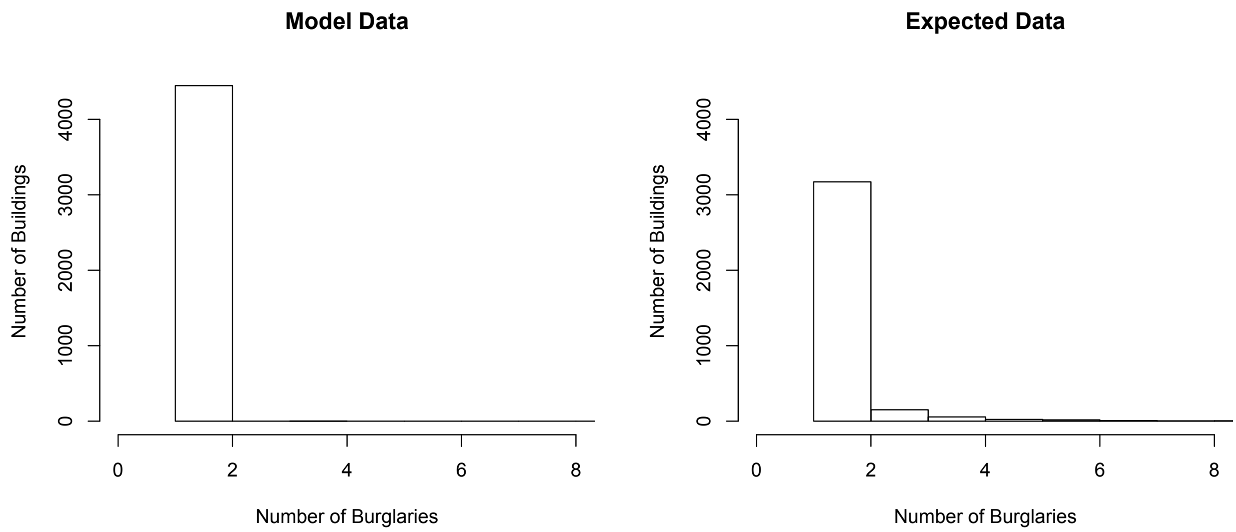

Figure 41: Frequency distribution showing the number of burglaries in individual buildings in real crime data.

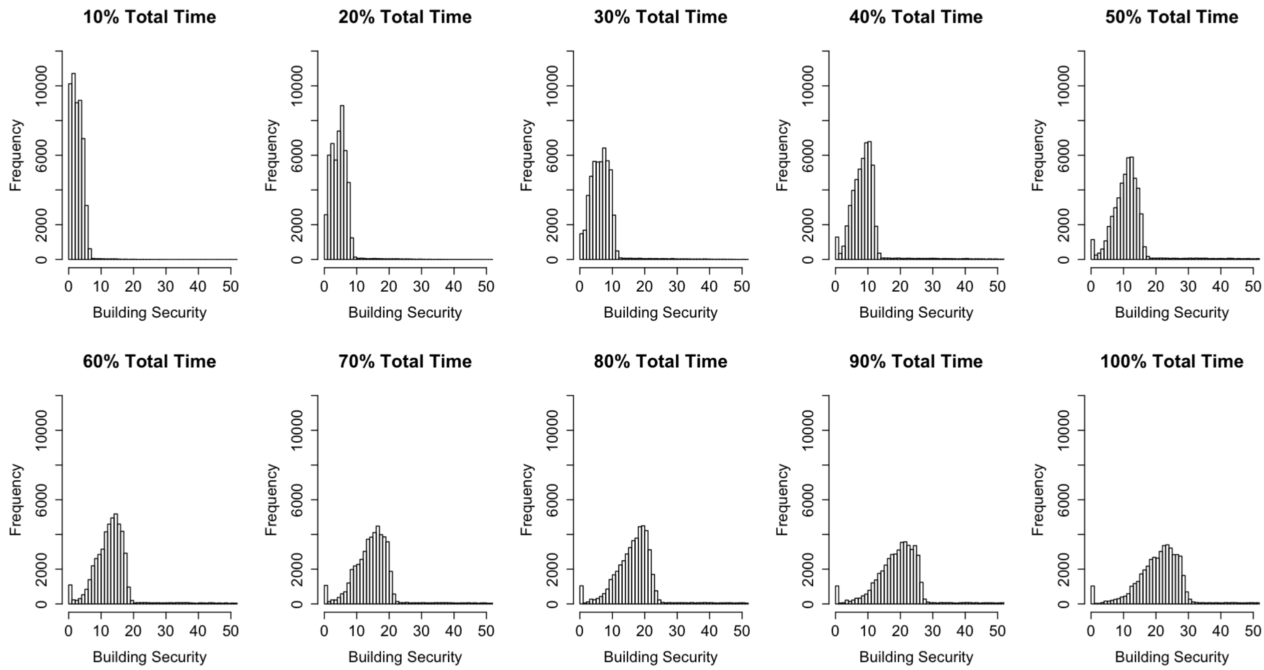

Figure 42: Frequency distribution of the security of all buildings in the EASEL area at different time points (the percentage of total time) in a Base 1 simulation.

absence of empirical evidence to base them on.

As with the previous simulation, Figure 43 illustrates the burglary densities produced by four randomly chosen models and Figure 44 graphs the L functions produced by all models. Both figures suggest that the simulation results are consistent; all results are similar. To determine whether or not the simulation reached equilibrium, Figure 45 maps the hotspots produced at various time points produced by a typical simulation. Again the simulation appears to reach equilibrium.

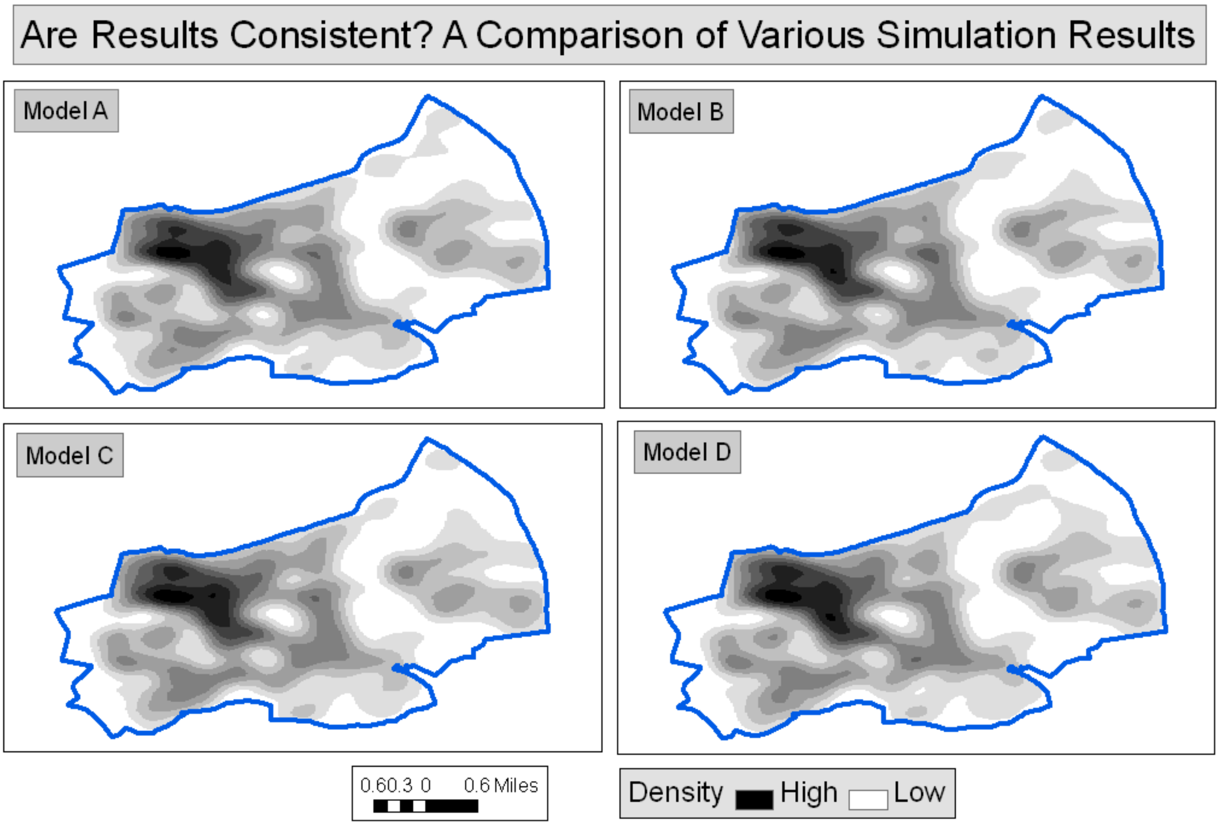

Figure 43: Map comparing different Base 2 models.

Finally, to compare the results to expected data, Figure 46 contrasts the density of a typical model to expected data and Figures 47 and 48 illustrate the overall error.

Table 26 compares the SRMSE and R^2 errors of the Base 1 and Base 2 scenarios at the resolutions mapped in Figure 47 and shows that the changes appear to have actually lowered the accuracy of the model slightly. More importantly, however, the graph of the L functions (Figure 44) shows that the burglaries are now more clustered than they were in the previous scenario and Figure 49 demonstrates that security levels no longer increase indefinitely.

Regardless of whether or not Base2 is a poorer simulation based on the distribution of burglary it is, however, a better simulation in the sense that it has brought the behaviour of the environment in line with our expectations. In Base1, house security increased indefinitely which is clearly inaccurate, this problem was solved by the changes made to the second scenario. Therefore if we assume that the environment is now behaving correctly in response to burglary (with respect to house security at least), there might be utility in continuing the calibration even though it appears more appropriate to disregard the Base2 changes because they did not

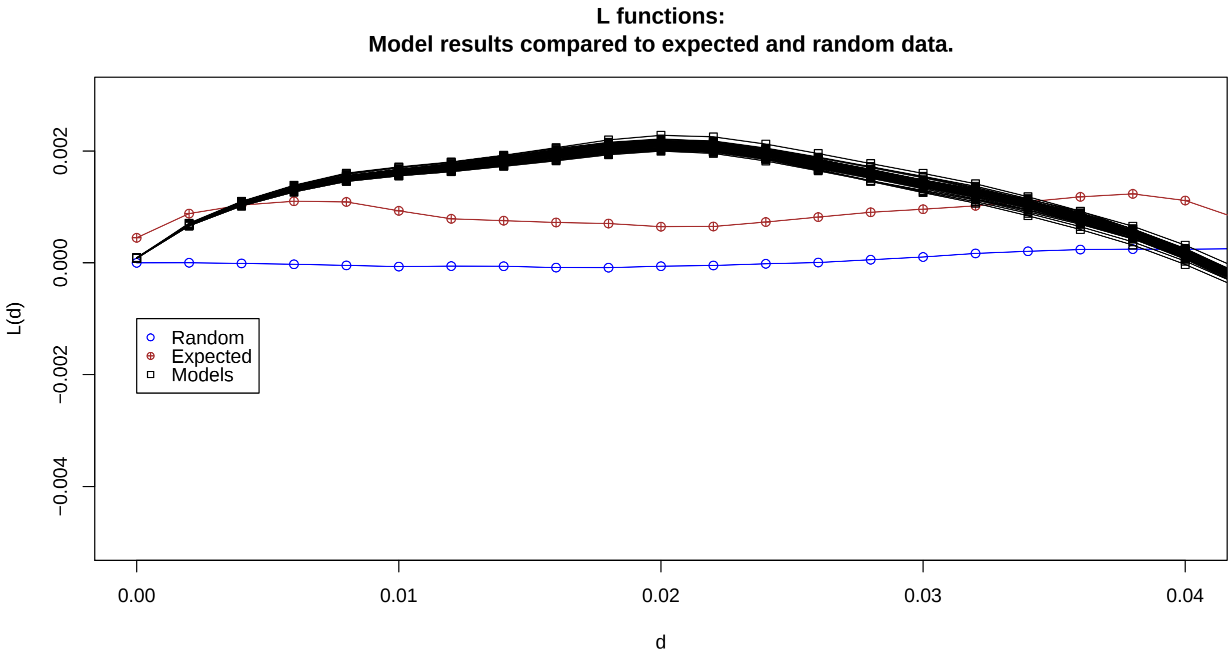

Figure 44: L Functions for a number of Base 2 models compared to expected data.

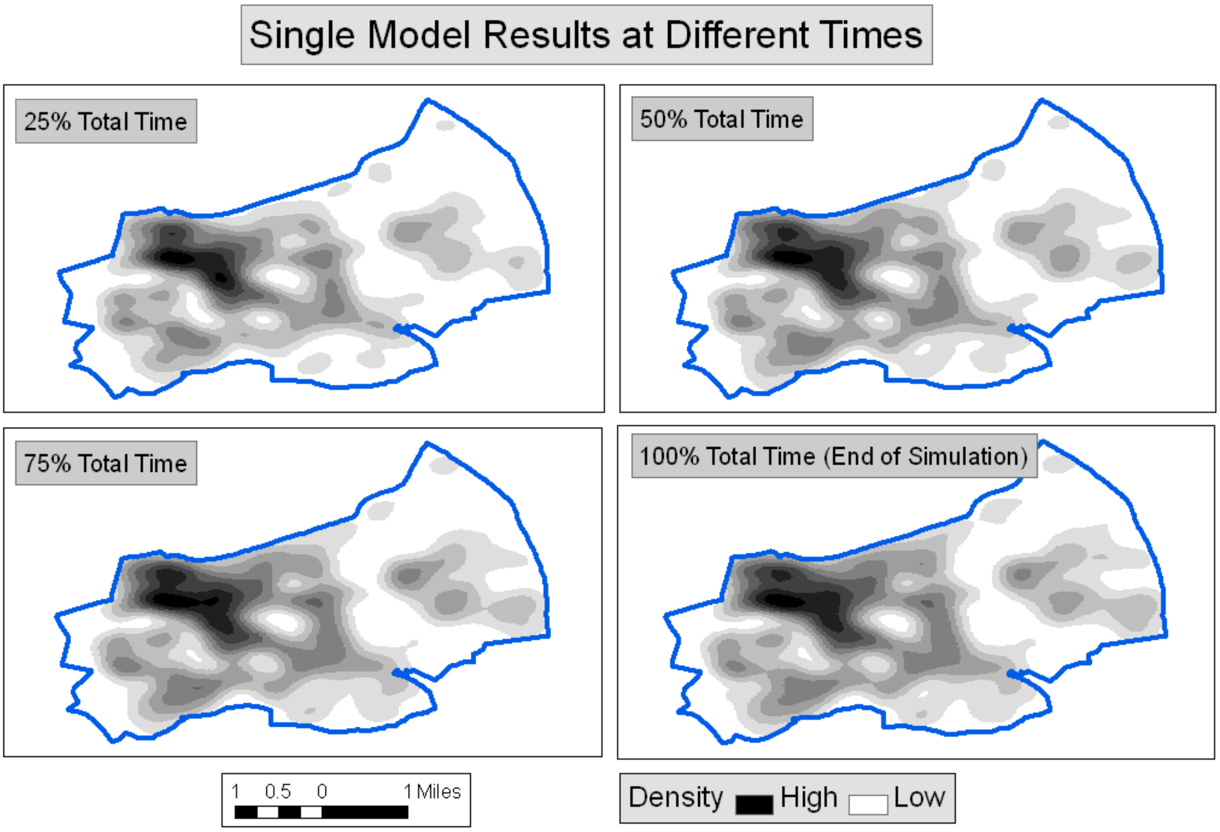

Figure 45: Map comparing hotspots produced by a single Base 2 model at different simulated times.

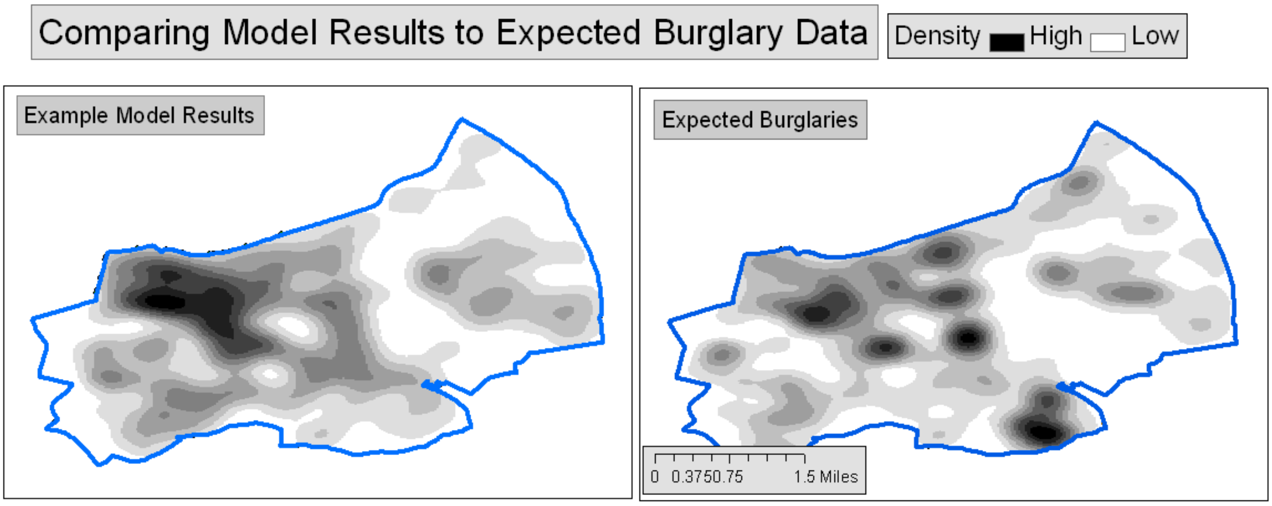

Figure 46: Map comparing Base 2 model results hotspots to expected data.

Table 26: Model errors at different cellular resolutions

		Base 1 Scenario		Base 2 Scenario	
Number of Cells	Resolution (Cell Area, km^2)	SRMSE	R2	SRMSE	R2
121	0.65	0.84	0.61	0.86	0.64
441	0.18	1.05	0.5	1.27	0.44

improve the model fitness.

4.4 Base 3 – Agents Dislike Security

The previous experiment changed the default behaviour so that the dynamics of household security changes after burglary were more appropriate. However, now it appears that burglary is *too* heavily spatially clustered (when compared to expected data). An alternative to experimenting further with security values is to change how the burglar agents perceive security precautions. At present, burglars view all household parameters equally (they do not consider security to be a greater deterrent than occupancy for example). This experiment will alter this so that, when deciding whether or not to burgle, agents give a greater weight to security precautions, i.e. they are more deterred by security than by other environmental factors. This can be accomplished by changing the *Security_W* parameter which influences how strongly household security deters a potential burglar. Deciding on a new weight, however, is non-trivial as there is no previous research that can help to quantify a new weight value. Previous experiments suggest that a value of 5 creates the appropriate behaviour: making crime moderately less heavily clustered. As with all previous results the following figures summarise the results.

To ensure that the model reaches equilibrium and is consistent across different simulations, Figure 50 maps burglary patterns for some typical models and Figure 51 supports this with a graph of the L functions of all models. Also, Figure 52 illustrates that the results have reached equilibrium.

To explore the burglary distributions spatially, Figure 53 compares the burglary rates to expected data.

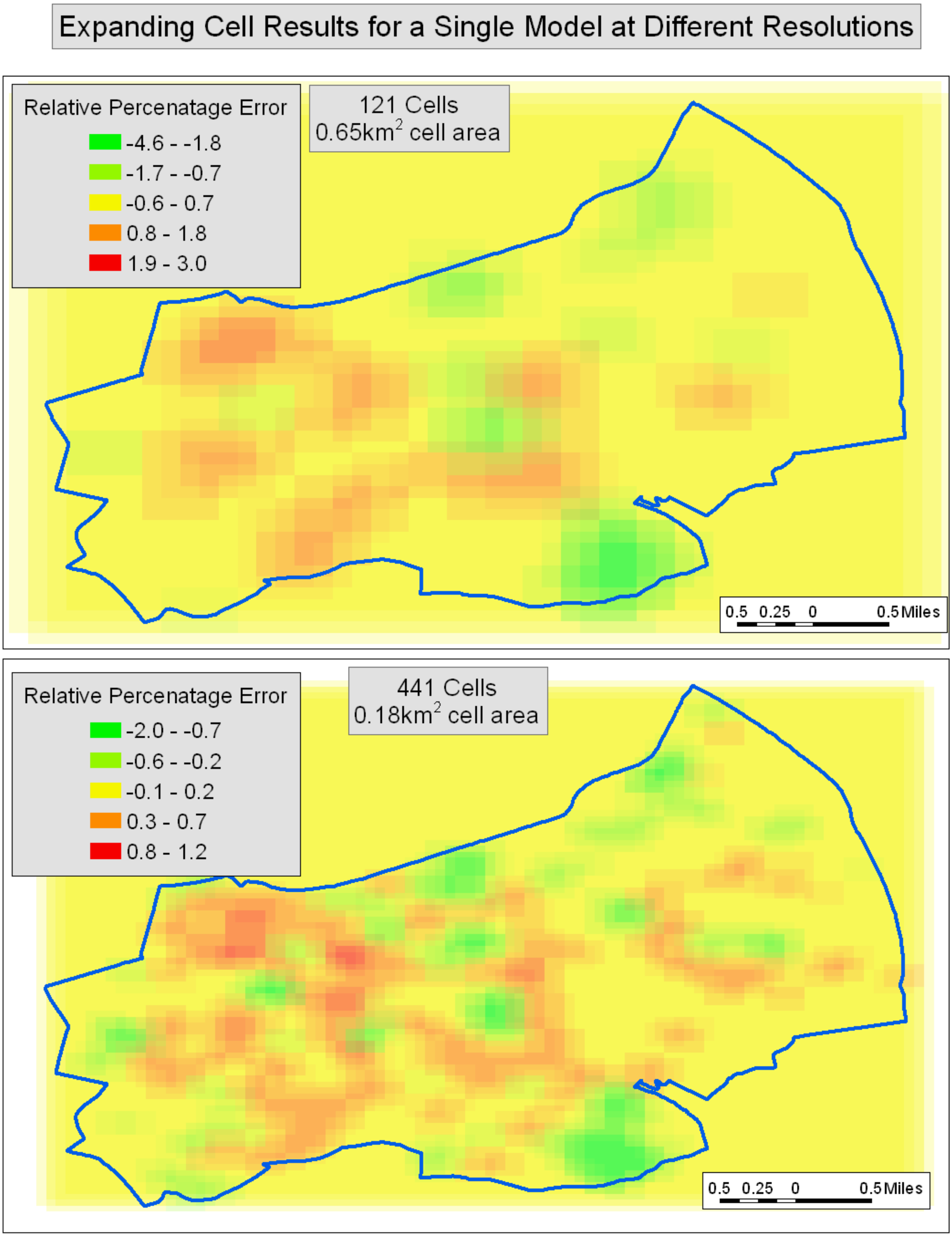

Figure 47: Expanding cell maps for a Base 2 Scenario

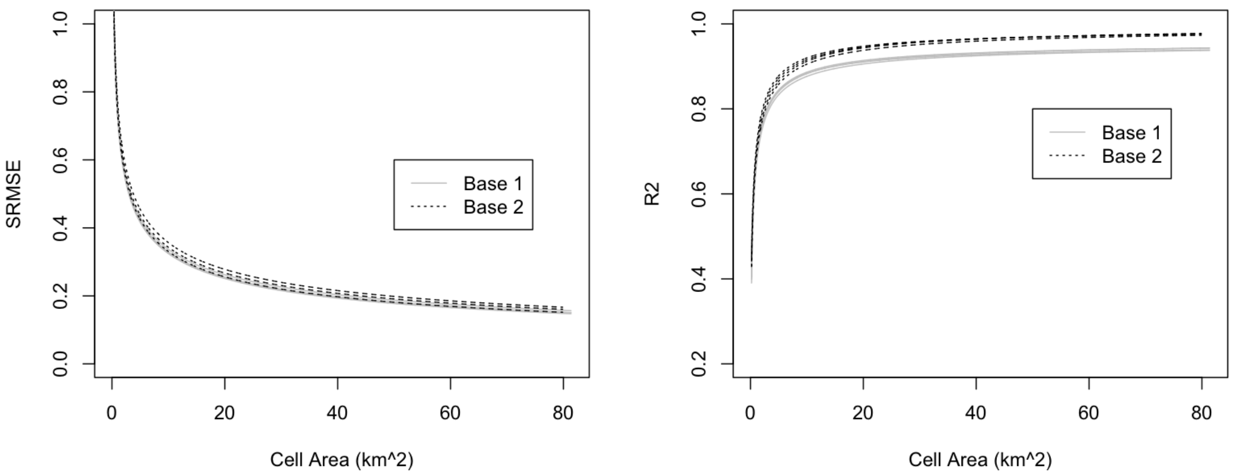

Figure 48: Expanding cell graphs for a Base 2 scenario compared to Base 1

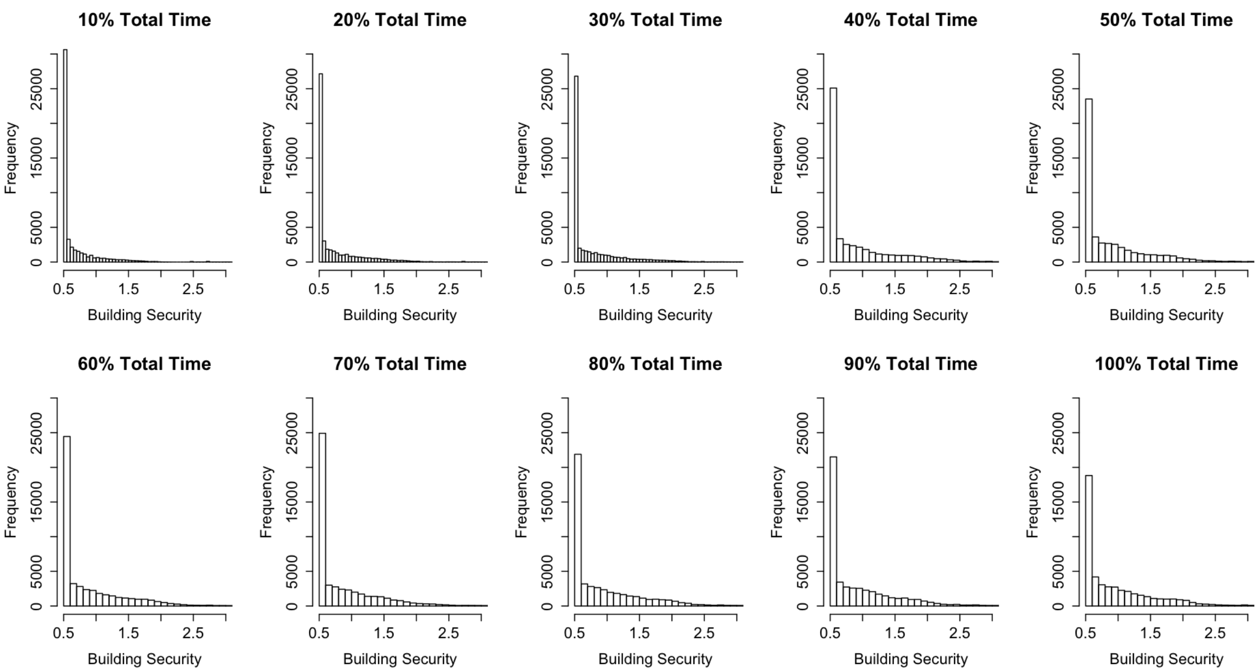

Figure 49: Frequency distribution of the security of all buildings at different time points in a Base 2 simulation.

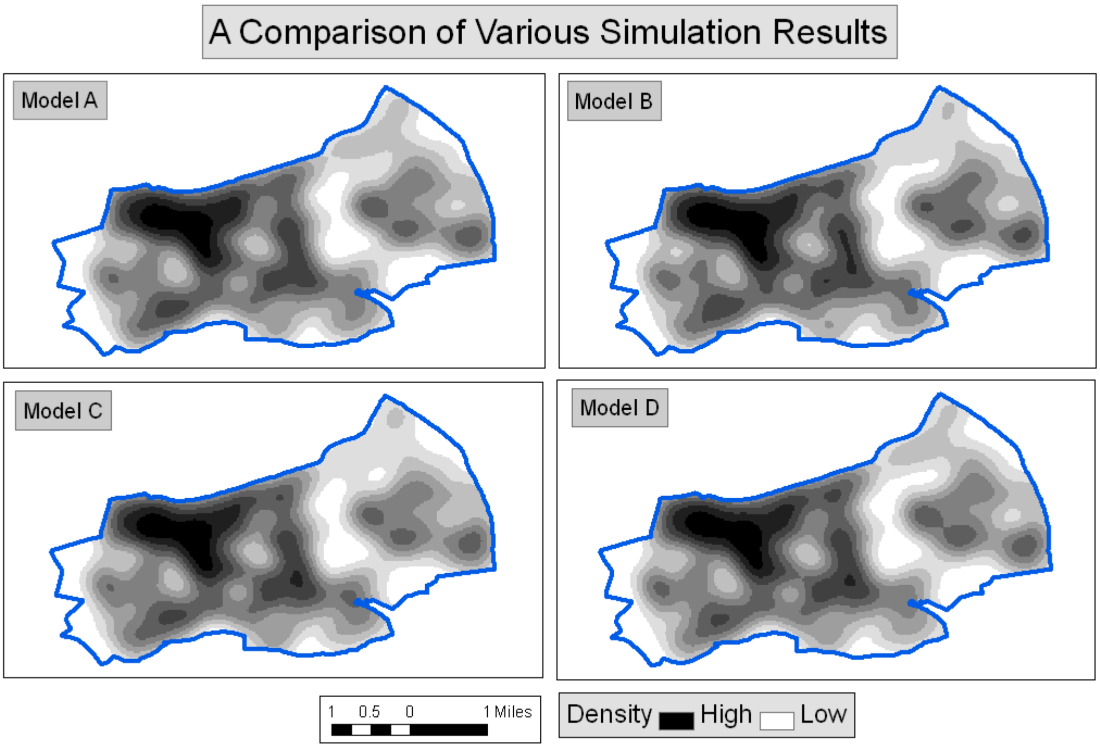

Figure 50: Map comparing different Base 3 models to show that they are consistent.

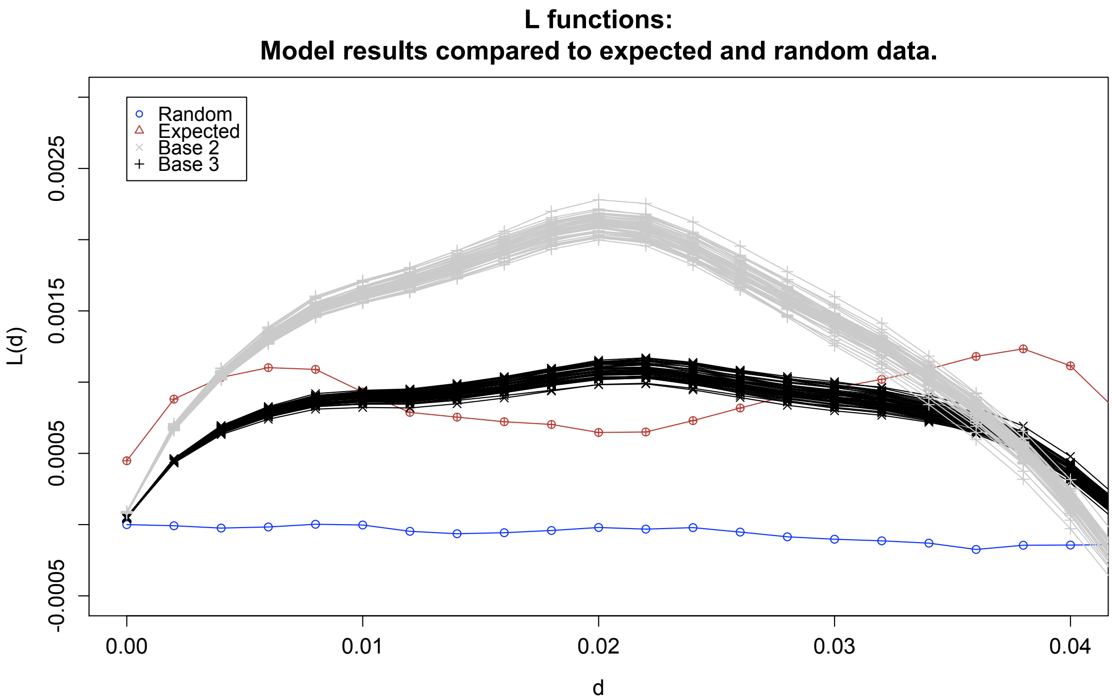

Figure 51: L Functions for a number of Base 3 models compared to Base 2 and expected data.

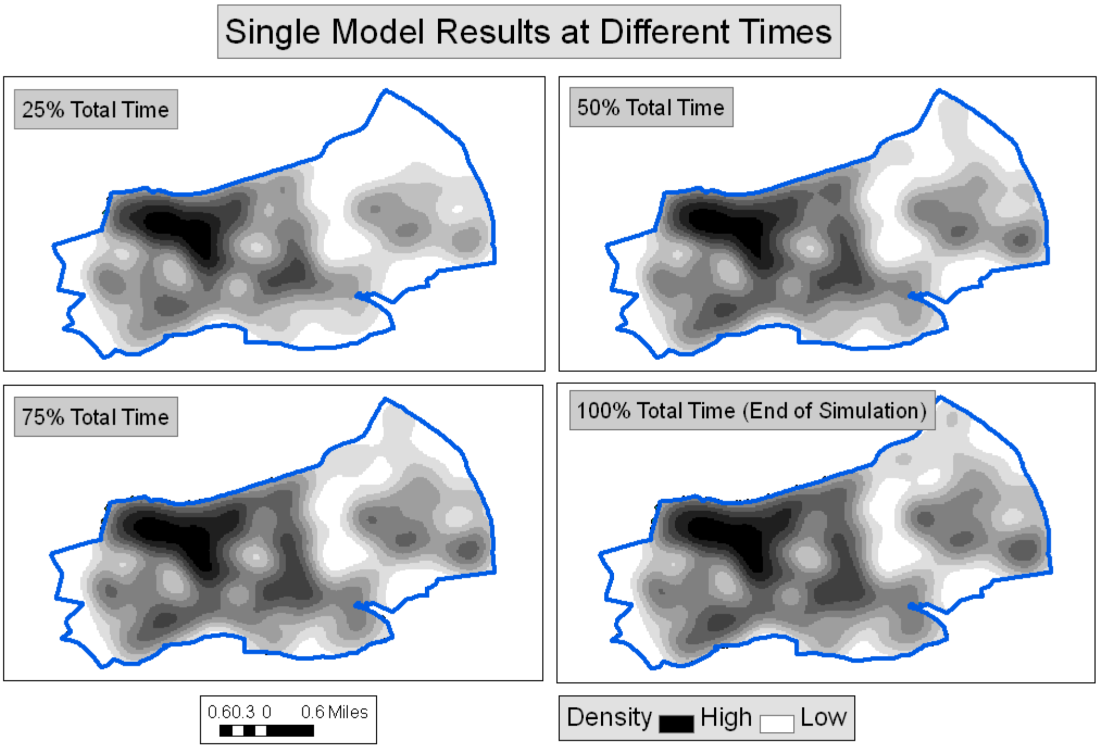

Figure 52: Map comparing hotspots produced by a single Base 3 model at different simulated times.

There are still some difference between the two data sets but they are clearly more similar than the Base 1 (Figure 35) and Base 2 (Figure 46) results.

The improvements of the Base 3 calibration are further illustrated by the expanding cell analysis; Figure 54 maps the model errors at two different resolutions. Comparing this map to the original (Base 1) expanding cell map (Figure 36) it is clear that a much larger area has similar crime levels (the thematic ranges are the same on both maps). Table 27 provides the global errors and the two mapped resolutions and Figure 55 graphs the errors. Again the global goodness-of-fit statistics support the visual analysis and suggest that the calibration has improved the accuracy of the model (with the exception of the 0.18km^2 resolution SRMSE measure which is *slightly* larger than the Base 1 equivalent).

Table 27: Model errors at different cellular resolutions. The errors that represent the greatest goodness-of-fit are bold.

Scenario	Number of Cells	Resolution (Cell Area, km^2)	SRMSE	R^2
Base 3 Scenario	121	0.65	**0.76**	**0.68**
	441	0.18	1.10	**0.54**
Base 2 Scenario	121	0.65	0.86	0.64
	441	0.18	1.27	0.44
Base 1 Scenario	121	0.65	0.84	0.61
	441	0.18	**1.05**	0.50

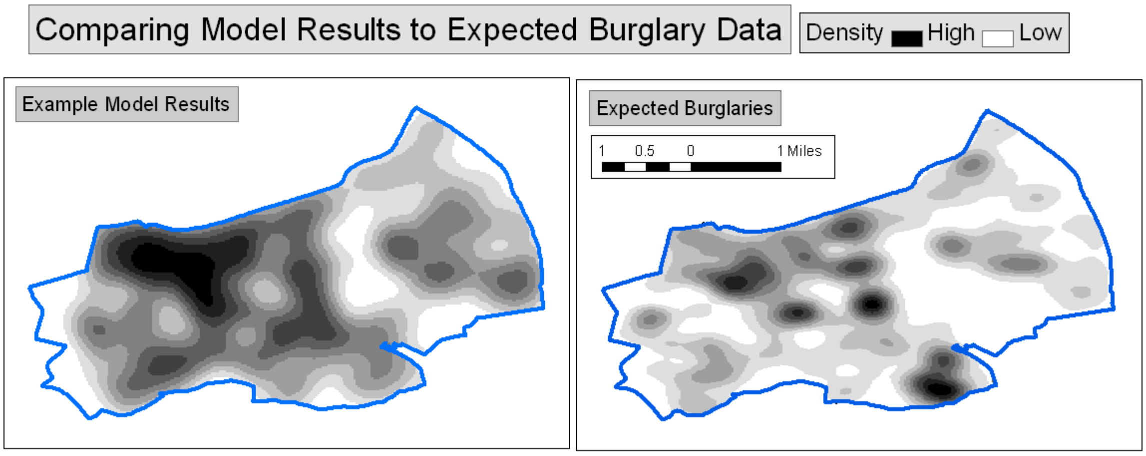

Figure 53: Map comparing model Base 3 hotspots to expected data.

4.5 Summary – Model Calibration

The calibration process has improved the global accuracy of the model (based on goodness-of-fit analysis) and has brought the spatial distribution of the resulting burglaries more in line with that of expected data. Also, it has generated a model that, depending on the type of aggregation used, is able to outperform a comparable regression model using the R^2 statistic to calculate error.

It was noted that calibrating the model using an automated routine, such as a genetic algorithm, was infeasible. Instead the model was calibrated manually. There are both advantages and disadvantages to this approach to calibration. An advantage, as illustrated by the second set of changes (Base 2), is that the researcher can use their knowledge of the model to perform changes that do not necessarily increase the fitness of a model and might be disregarded by an automatic routine. A disadvantage, however, is that the parameter space is not explored nearly as comprehensively as it would be by a computer algorithm. There are therefore various routes which could potentially be taken in order to calibrate the model further. The behaviour of the burglars could be improved in order to better simulate crime rates in the Halton Moor area which was very poorly simulated. It would also be interesting to analyse the SRMSE and R^2 error statistics in more detail to establish why they are occasionally in disagreement regarding the most accurate result. Or the error graphs could be analysed to estimate the resolution at which the model is able to make reliable predictions to. However, at this stage the model is deemed to perform adequately on the calibration dataset and will now be used to make predictions. Model calibration will continue to be an area for further research in the future.

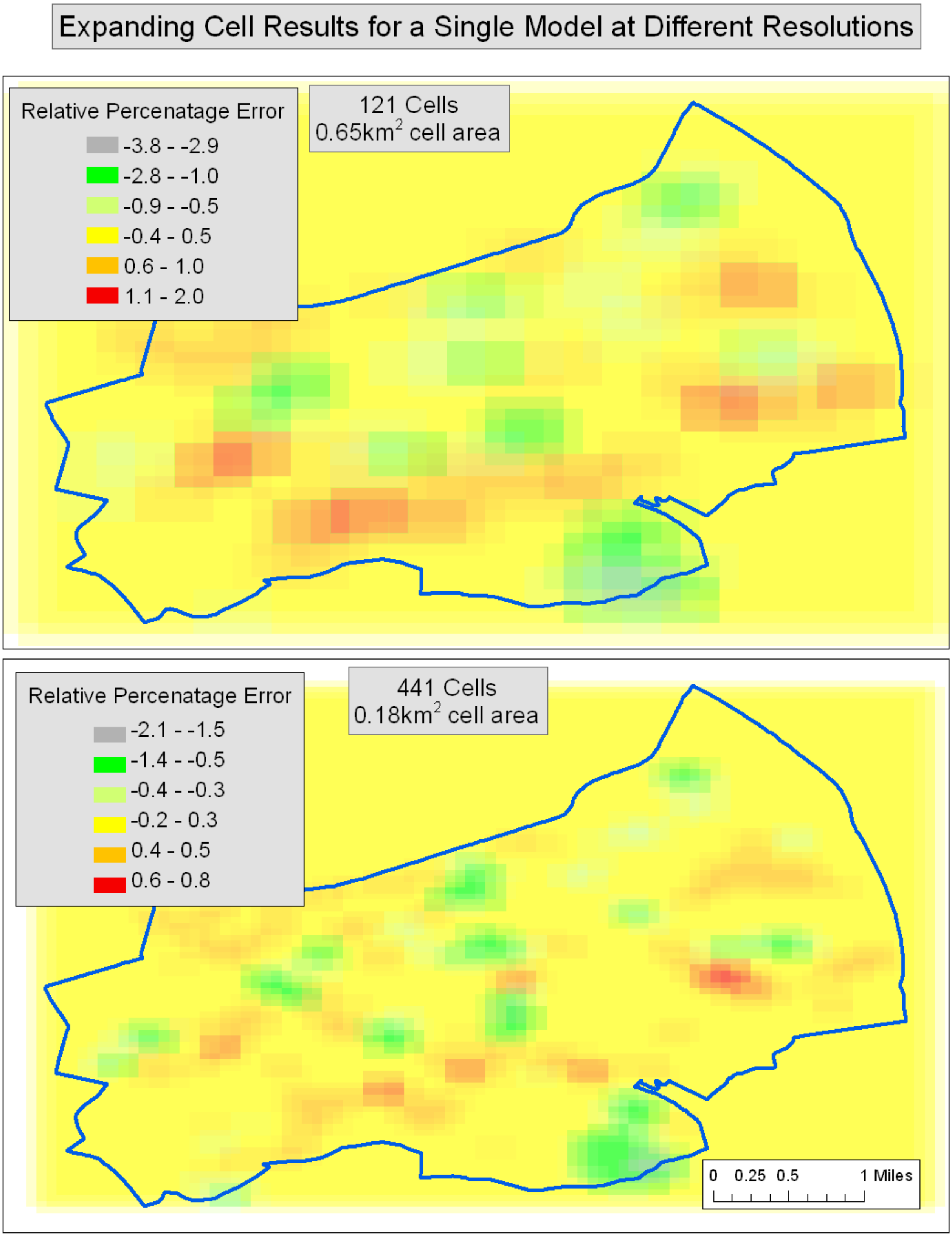

Figure 54: Expanding cell maps for a Base 3 Scenario.

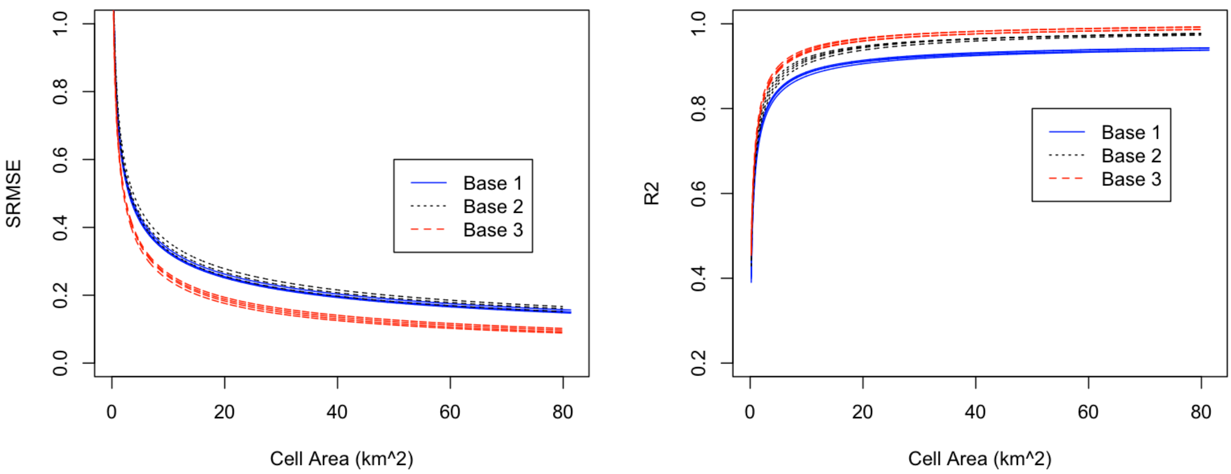

Figure 55: Comparing all Base results at different resolutions.

5 Validation

The previous sections have verified that the model works as expected (Section 3) and calibrated it to reflect the main trends and behavioural processes using 2001 field data (Section 4). The final task before the model can be used to perform experiments is to ensure that the model has not been *over-fitted* to the available data and is able to forecast in situations outside that on which it was calibrated. This process is often termed validation.

Validating this research, however, poses some problems. The most significant is that most of the data used to create the virtual environment are based on the 2001 UK census and no data for any other years is available. Therefore validating the model on subsequent years will be inherently inaccurate because the environment cannot be updated to reflect the changes that will have occurred to the demographics since 2001. Furthermore, offender data are only available up to 2004 so for any other years there is no way to seed the offender home locations. Therefore the model will be validated on 2004 data. This is the most suitable year because it is close enough to 2001 so that the demographics of the EASEL area should be similar but far enough from 2001 so that crime patterns will be different.

5.1 Preparing the Data Crime Data

For consistency with the 2001 data, the 2004 victim and offender data sets would ideally cover the period 1st April 2003 to 31st March 2005 to make the data more dense. However, as offender data is not available past 31st March 2004 the period will be 1st April 2003 to 31st March 2004 instead. Figure 56 illustrates the crime hotspots generated from the victim data

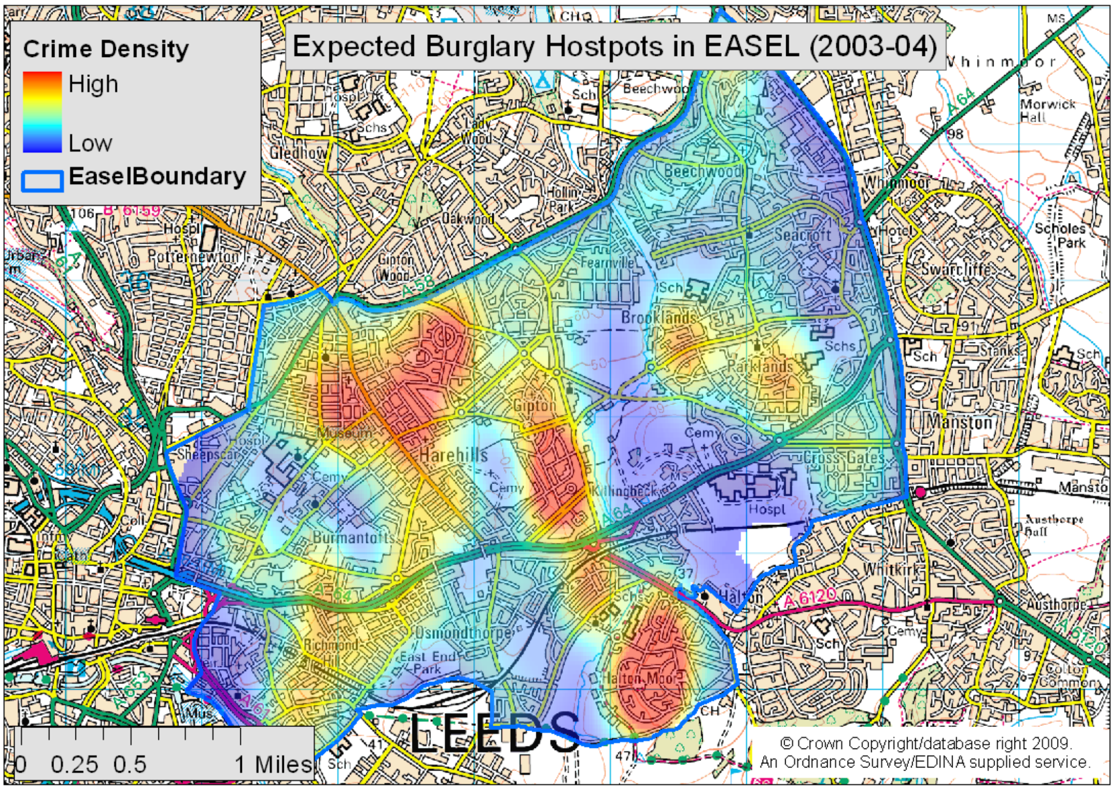

Figure 56: EASEL crime hotspots in the 2004 data.

5.2 Validation Results

The model was configured using the final calibration configuration (Base 3) and, as with all other experiments, was run for 30 days and executed 50 times. As there is no 2004 environmental data to use, the only difference in input data are the locations of the offenders. As with other experiments, the results are determined to be consistent as follows:

- Figure 57 maps burglary patterns for some typical models to demonstrate that the results are consistent.
- Figure 58 illustrates that the results have reached equilibrium.
- Figure 59 compares the burglary rates to expected data.

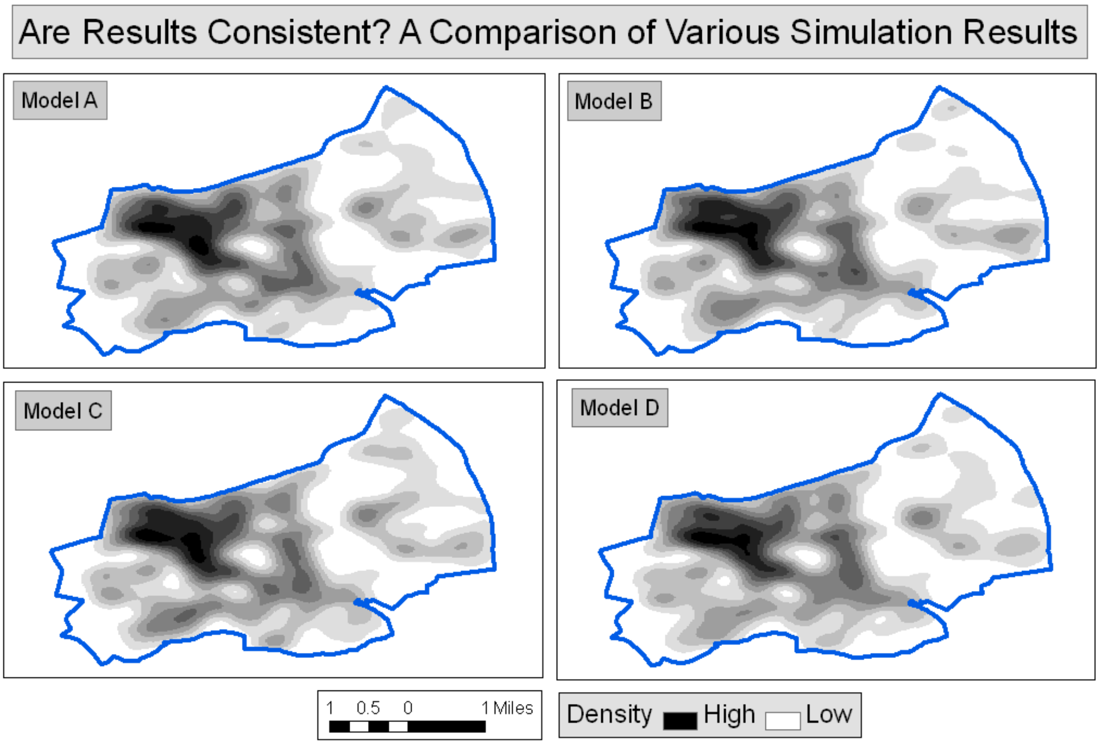

Figure 57: Map comparing different Validation models to show that they are consistent.

Figure 59 compares the burglary rates to expected data. There are clear differences; in particular crime in Halton Moor has been significantly under predicted again. This is to be expected however because the motivations that drive agent behaviour have not been changed. As Section 4.2 discussed, it is possible that the burglary hotspot in Halton Moor is not a result of burgling for financial gain (as suggested by the crime literature) but rather for intimidation or other reasons (Safer Leeds, personal communication). Therefore, the model (in its current configuration) is not able to account for this hotspot. Interesting future work would be to experiment with the behaviour of offender agents to try to replicate the hotspot more closely but this is beyond the scope of this research.

Figure 60 graphs the expanding cell errors of all models including this validation scenario. Ideally, the validation experiment should produce a similar fitness to the final (Base 3) calibration experiments. Unfortunately, the fitness of the validation experiment, measured using R^2 and SRMSE, is lower than the calibration.

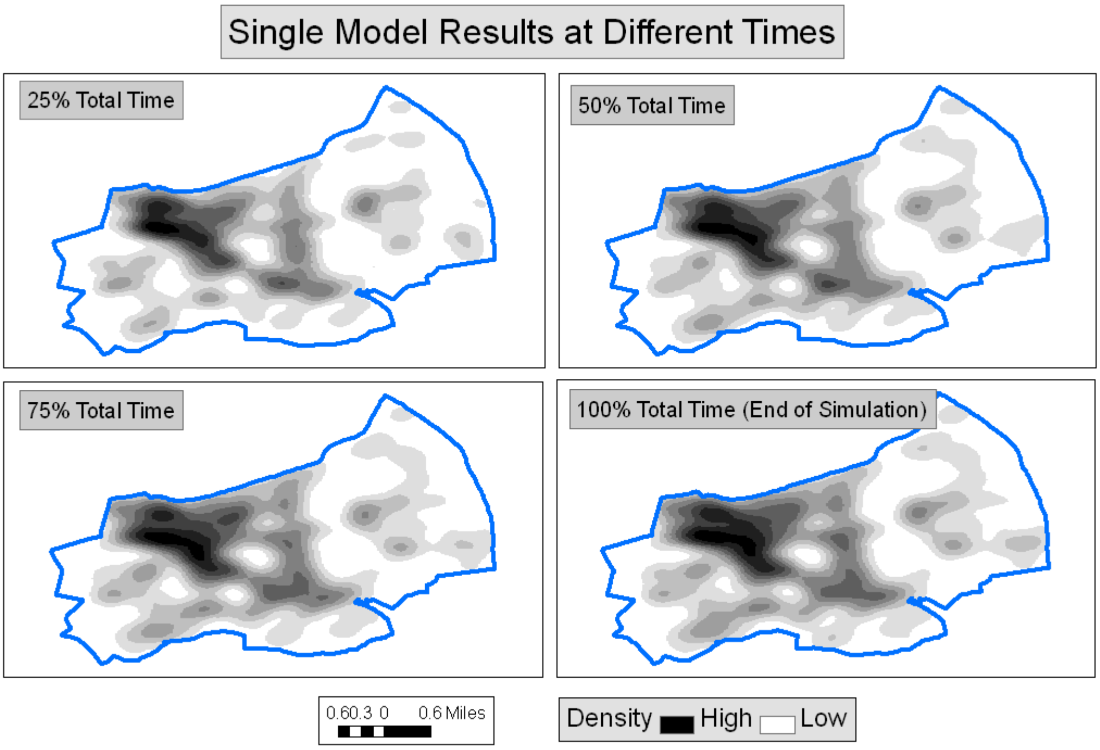

Figure 58: Map comparing hotspots produced by a single Validation model at different simulated times.

Usually this would suggest that the model has been over-fitted to the 2001 field data. However, at the start of this section it was noted that no environmental data for 2004 were available (the UK census only occurs every decade). Therefore it is probable that the poor performance of the validation model is a result of inadequate input data, rather than a being over-fitted to 2001 data.

5.3 Validation – Summary

The validation process suggests that the model does not perform as well as it did for the calibration ("base") experiments. Under normal circumstances this would indicate that the model was over-fitted to the calibration data and needed to be improved to make it useful in other situations. However, in this case there are a number of factors that might lead to poorer than expected validation results. The most likely reason is that there was no updated census data to use as input for the model. Others include missing model parameters (e.g. crime reduction scenarios in 2004 have changed the way in which people respond to burglary, rendering some of the calibration changes unrealistic) or poor assumptions about how offenders respond to environmental cues. Unfortunately it is difficult to verify whether or not these are the case in the absence of accurate data on which to validate the model.

The validation process is not without its uses, however. It has demonstrated the risks associated with applying the model to scenarios for which sufficient data are lacking. Therefore when the next chapter explores the power of the model as a forecasting tool it will do so with care.

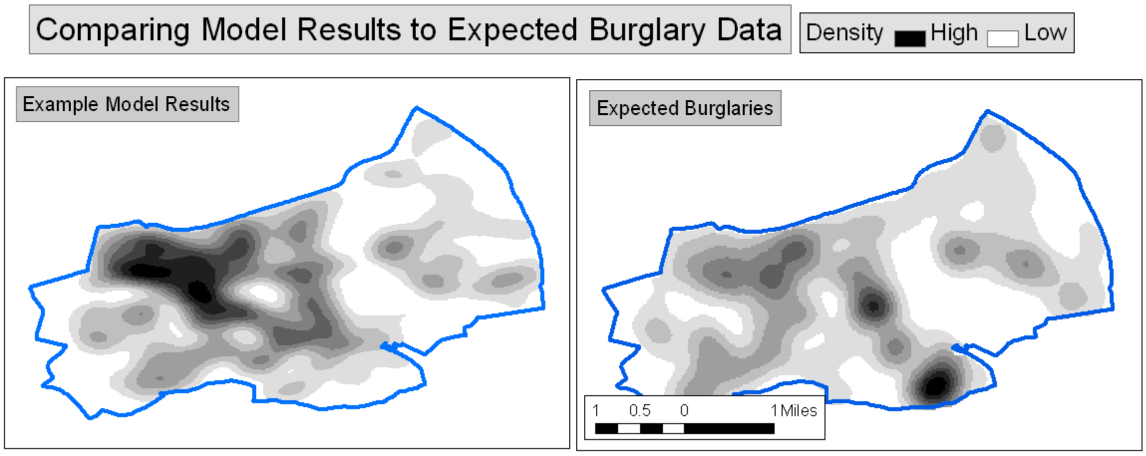

Figure 59: Map comparing validation result hotspots to 2004 expected data.

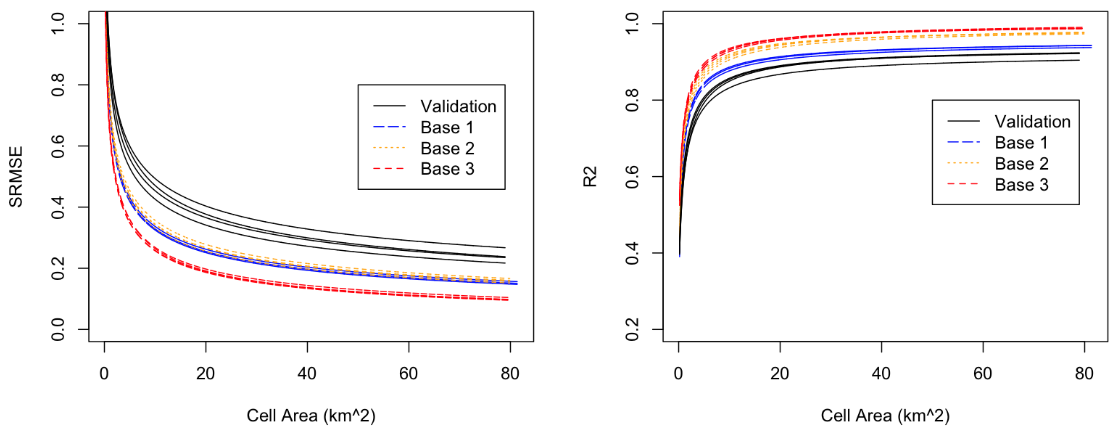

Figure 60: Comparing the fitness of the validation results (using 2004 crime field data) to base results (using 2001 field data) at different resolutions.

6 Evaluating the Model – Summary

The burglary model developed for this research uses a very large number of parameters and is accordingly extremely difficult to evaluate. This paper has discussed some of the repercussions of this and experimented with all the important parameters in order to first verify that the model is free from errors and performs as it should do, then to calibrate it and finally to validate it using an alternative data set.

Section 2 began with an in-depth review of methods that can be used to compare point patterns. It was decided that no method in isolation was sufficient at providing a sound process for evaluating the model. Therefore a detailed analysis framework was devised that would allow for model results to be evaluated both by comparing the spatial locations of points to expected data (the "expanding cell" method) as well as comparing the spatial structure of the data through the use of L functions. In combination, these methods provide a complete and robust means of evaluating the accuracy of the model at different resolutions. This method has significant advantages over other methods, such as avoiding the requirement of aggregating up to an area boundary before applying goodness-of-fit tests.

The main evaluation began with Section 3 which verified that the model had been programmed correctly. It was decided that re-implementing the model in a different programming language was infeasible so, instead, the model was executed on different types of environment that increased in complexity gradually. This novel approach also had the advantage that it allowed for sensitivity tests to be performed accurately as environmental complexity could be limited. This is in response to a concern among some environmental criminologists (e.g. Elffers and van Baal (2008)) that models with complicated environments can detract from understanding the underlying processes which is the main advantage of this type of model.

Having verified the model, the final stages were to calibrate and validate it. Section 4 outlined the process of calibration. The advantages of using multiple methods to compare simulated and result data became clear when the model began to perform more accurately, even though this initially led to a drop in the global goodness-of-fit. Finally Section 5 attempted to validate the model by executing a new scenario using crime data from an alternative year (2004). The validation results were poorer than the calibration results which might suggest that the model has been over-fitted to the calibration data. However, there is insufficient input data to build a reliable virtual city that represents 2004 and it is likely that this accounts for the poor validation performance.

References

Axelrod, R. (1997). Advancing the art of simulation in the social sciences. In R. Conte, R. Hegselmann, and P. Terna (Eds.), *Simulating Social Phenomena*, pp. 21–40. Berlin: Springer-Verlag.

Axtell, R. (2000). Why agents? On the varied motivations for agent computing in the social science. Center on Social and Economic Dynamics Working Paper No. 17, available at `http://www.brookings.edu/es/dynamics/papers/agents/agents.htm` [accessed - January 2007].

Axtell, R., R. Axelrod, J. M. Epstein, and M. D. Cohen (1996). Aligning simulation models: A case study and results. *Computational and Mathematical Organization Theory 1*(2), 123–141.

Bailey, T. and T. Gatrell (1995). *Interactive Spatial Data Analysis*. Prentice Hal.

Baldwin, J. and A. E. Bottoms (1976). *The Urban Criminal: A Study in Sheffield*. London: Tavistock Publications.

Bennell, C., B. Snook, P. J. Taylor, S. Corey, and J. Keyton (2007). Its no riddle, choose the middle: The effect of number of crimes and topographical detail on police officer predictions of serial burglarshome locations. *Criminal Justice And Behavior 34*(1), 119–132.

Bernasco, W. and P. Nieuwbeerta (2005). How do residential burglars select target areas? *British Journal of Criminology 45*(3), 296–315.

Bottoms, A., A. Claytor, and P. Wiles (1992). Housing markets and residential community crime careers: A case study from Sheffield. In N. Fyfe, D. Evans, and D. Herbert (Eds.), *Crime, Policing and Place – Essays in Environmental Criminology*, Chapter 7, pp. 94–116. London: Routledge.

Brantingham, P. J. and P. L. Brantingham (1984). *Patterns in Crime*. Macmillan USA.

Brantingham, P. L. and P. J. Brantingham (1981). Mobility, notoriety, and crime: A study in the crime patterns of urban nodal points. *Journal of Environmental Systems 11*(1), 89–99.

Brown, D. (2005). Agent-based models. In H. Geist (Ed.), *Our Earth's Changing Land: An Encyclopedia of Land-Use and Land-Cover Change*, pp. 7–13. Greenwood Publishing Group.

Casti, J. (1997). *Would-Be-Worlds: How Simulation is Changing the Frontiers of Science*. New York: John Wiley and Sons.

Castle, C. J. E. and A. T. Crooks (2006). Principles and concepts of agent-based modelling for developing geospatial simulations. Available online at `http://eprints.ucl.ac.uk/archive/00003342/01/3342.pdf`. UCL Working Papers Series, Paper 110, Centre For Advanced Spatial Analysis, University College London.

Chainey, S. (2009a, 7th-8th May). Advanced hotspot analysis. Presentation at the 7th National Crime Mapping Conference, Manchester, UK, available at `http://www.cscs.ucl.ac.uk/events-1/cmc/CMCClass5DChainey.pdf`.

Chainey, S. (2009b, 7th-8th May). Understanding hotspots. Presentation at the 7th National Crime Mapping Conference, Manchester, UK, available at `http://www.cscs.ucl.ac.uk/events-1/cmc/CMCClass3CChainey.pdf`.

Chainey, S. and J. Ratcliffe (2005). *GIS and Crime Mapping*. John Wiley & Sons.

Clark, P. J. and F. C. Evans (1954). Distance to nearest neighbour as a measure of spatial relationships in populations. *Ecology 35*, 445–453.

Cohen, L. and M. Felson (1979). Social change and crime rate trends: A routine activity approach. *American Sociological Review 44*, 588–608.

Costanza, R. (1989). Model goodness of fit: A multiple resolution procedure. *Ecological Modelling 47*, 199–215.

Eck, J. E., S. Chainey, J. G. Cameron, M. Leitner, and R. E. Wilson (2005). *Mapping Crime: Understanding Hot Spots*. Washington, DC: U.S. Department of Justice, Office of Justice Programs.

Elffers, H. and P. van Baal (2008). Realistic spatial backcloth is not that important in agent based simulation: An illustration from simulating perceptual deterrence. In L. Liu and J. Eck (Eds.), *Artificial Crime Analysis Systems: Using Computer Simulations and Geographic Information Systems*, Chapter 2, pp. 19–34. IGI Global.

ESRI (2006). *ArcGIS 9.2 Desktop Help*. Redlands, USA: ESRI.

Fotheringham, S. and P. Rogerson (Eds.) (1994). *Spatial Analysis and GIS*. Taylor and Francis.

Getis, A. and J. Ord (1992). The analysis of spatial association by use of distance statistics. *Geographical Analysis 24*, 189–206.

Getis, A. and J. K. Ord (1996). Local spatial statistics: an overview. In P. Longley and M. Batty (Eds.), *Spatial analysis: modelling in a GIS environment*, pp. 261–277. Cambridge: Geoinformation International.

Gilbert, N. and P. Terna (2000). How to build and use agent-based models in social science. *Mind & Society 1*, 57–72.

Hagen, A. (2003). Fuzzy set approach to assessing similarity of categorical maps. *International Journal of Geographical Information Science 17*(3), 235–249.

Hagen-Zanker, A., G. Engelen, J. Hurkens, R. Vanhout, and I. Uljee (2005). *Map Comparison Kit 3 User Manual*. Maastricht, The Netherlands: Research Institute for Knowledge Systems (RIKS bv).

Harland, K. (2008). *Journey to Learn: Geographical Mobility and Education Provision*. Ph. D. thesis, School of Geography, University of Leeds, Leeds LS2 9JT, UK.

Heppenstall, A., A. Evans, and M. Birkin (2006). Application of multi-agent systems to modelling a dynamic, locally interacting retail market. *Journal of Artificial Societies and Social Simulation 9*(3).

Heppenstall, A. J. (2004). *Application of Hybrid Intelligent Agents to Modelling a Dynamic, Locally Interacting Retail Market*. Ph. D. thesis, School of Geography, University of Leeds, Leeds LS2 9JT, UK.

Knudsen, D. C. and A. S. Fotheringham (1986). Matrix comparison, goodness-of-fit, and spatial interaction modeling. *International Regional Science Review 10*, 127–147.

Kongmuang, C. (2006). *Modelling Crime: A Spatial Microsimulation Approach*. Ph. D. thesis, School of Geography, University of Leeds, Leeds LS2 9JT, UK.

Kuhnert, M., A. Voinov, and R. Seppelt (2005, August). Comparing raster map comparison algorithms for spatial modeling and analysis. *Photogrammetric Engineering & Remote Sensing 71*(8), 975–984.

Levine, N. (2006). Crime mapping and the crimestat program. *Geographical Analysis 38*, 41–56.

Levine, N. (Ed.) (2009). *CrimeStat Version 3.2 Manual*. Houston, TX: Ned Levine and Associates.

Malleson, N. (2006). An agent-based model of burglary in leeds. Master's thesis, University of Leeds, School of Computing, Leeds LS2 9JT, UK. Available on-line at `http://www.geog.leeds.ac.uk/fileadmin/downloads/school/people/postgrads/n.malleson/mscproj.pdf`.

Malleson, N. (2008). Simulating burglary: what does the future hold for leeds? *The Yorkshire and Humber Regional Review 18*(2).

Malleson, N. and P. Brantingham (2009). Prototype burglary simulations for crime reduction and forecasting. *Crime Patterns and Analysis 2*(1).

Malleson, N., A. Evans, and T. Jenkins (2009). An agent-based model of burglary. *Environment and Planning B: Planning and Design 36*, 1103–1123.

Malleson, N., A. Heppenstall, and L. See (2009a). Crime reduction through simulation: An agent-based model of burglary. *Computers, Environment and Urban Systems In Press*.

Malleson, N., A. Heppenstall, and L. See (2009b, December). Simulating burglary with an agent-based model. Working Paper 09/3, School of Geography, University of Leeds, December 2009. Available online: `http://www.geog.leeds.ac.uk/fileadmin/downloads/school/research/wpapers/09-03.pdf`.

Metropolitan Police (2009). Metropolitan police crime mapping. `http://maps.met.police.uk/` [accessed – Oct 2009].

O'Sullivan, D. (2004, March 5–8). Complexity science and human geography. In *Transactions of the Institute of British Geographers*, Volume 29, pp. 282–295. Royal Geographical Society (with the Institute of British Geographers).

O'Sullivan, D. and D. J. Unwin (2003). *Geographic Information Analysis*. John Wiley and Sons.

Ripley, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society B 39*(2), 172–212.

Shepherd, P. J. (2006). *Neighbourhood profiling and classification for community safety*. Ph. D. thesis, School of Geography, University of Leeds, Leeds, UK.

Silverman, B. (1986). *Density estimation: for statistics and data analysis*. London: Chapman and Hall.

Snook, B. (2004). Individual differences in distance travelled by serial burglars. *Journal of Investigative Psychology and Offender Profiling 1*, 53–66.

van Nes, A. (2006). The burglar as a space explorer in his own neighborhood. In U. Mander, C. Brebbia, and E. Tiezzi (Eds.), *The Sustainable City IV. Urban regeneration and Sustainability*. Wessex, UK: WIT Press.

Vickers, D. and P. Rees (2006). Introducing the national classification of census output areas. *Population Trends 125*, 15–29.

Voas, D. and P. Williamson (2000). An evaluation of the combinatorial optimisation approach to the creation of synthetic microdata. *International Journal of Population Geography 6*, 349–366.

Wiles, P. and A. Costello (2000). *The 'road to nowhere': the evidence for travelling criminals.* Home Office Research Study 207. London: Home Office.

Wrigley, N. (1995). Revisiting the modifiable areal unit problem and the ecological fallacy. In A. D. Cliff, P. Gould, A. Hoare, and N. Thrift (Eds.), *Diffusing Geography: Essays for Peter Haggett*. Blackwell.