A Crime Aggregation Model On Street Networks (CAMOSNet)

Vincent T. Nguyen[1], Patricia L. Brantingham[1], Razvan C. Fetecau[1]

Abstract

In this article, we develop a mathematical framework to model crime and crime concentrations on a city road network. The model proposed is an advancement to similar frameworks inspired by a model introduced by Short et al. (2008). A significant modification introduced in our model is the use of spectral graph theory to represent the road network and to simulate diffusion throughout the network. The techniques discussed are tested in a simulation model of crime applied to the city of Vancouver, BC, Canada. The simulations presented are based off of empirical data of crime in Vancouver along with its street network. Results of the simulations present crime patterns that are consistent with crime patterns observed in the city.

1. Introduction

The significance of spatial-temporal criminology has culminated in research from disciplines in criminology, sociology, geography, psychology, computing science and mathematics. Simulating crime and modeling the dynamics of crime concentrations are significant areas of research in computing science and mathematics relating to spatial-temporal criminology. The value of research in these areas is in the predictive nature of modeling and simulations. How accurate a model or simulation is in predicting crime and crime concentrations is powerful in reducing the rate and spread of crime throughout a city.

In recent years, several mathematical models have been developed with the intention of modeling the dynamics of crime concentrations of hot spots through mathematical diffusion (Chaturapruek et al., 2013; Pan et al., 2018; Short et al., 2008). These models are all inspired by a framework proposed by Short et al. (2008) and we will refer to this collection of papers as the diffusion papers for their application of diffusion in modeling crime hot spots.

In our research, we are interested in studying the dynamics of these hotspots on a city's road network. The model we present in this article is an advancement to the models developed in the diffusion papers. It draws from similar techniques used in these papers including framing crime hot spots from the perspective of mathematical diffusion.

Although our model is similar to that of the diffusion papers, a key difference is that our model is applied directly to a city road network. In contrast, the diffusion papers model crime concentrations on both one and two dimensional lattices. In this article, we test our techniques by replicating crime in Vancouver, BC, Canada in a simulation model of crime. The methodology used in this simulation model is discussed and an error analysis between our results and the city's crime data is presented.

[1] Simon Fraser University, Burnaby, BC, Canada

2. Preliminaries

2.1 Network Theory

The underlying structure of our mathematical model is of course, a city road network. The study of networks has grown rapidly in the past several decades; especially with the advent of machine learning and social networks. The mathematical field in which discrete mathematicians study networks is called graph theory (West, 2000). We will, however, refer to graph theory as network theory in this article. A network in this field consists of three components: a set of points, the edges between them, and a relation defining the connectivity of the points. We note that points are also referred to as vertices or nodes and we say that two nodes are adjacent if there exists an edge between them. Additionally, we define the degree of a node to be the number of nodes adjacent to our given node.

One method network theorists use to describe the connections between nodes is called an *adjacency matrix* which we denote as $\mathcal{A}$. Given a graph G with m nodes, we number each node uniquely with the numbers $1, \dots, m$ and create an $m \times m$ matrix with these nodes. The element of the i^{th} row j^{th} column, which we denote as $\mathcal{A}_{ij}$, is the same as the element of the j^{th} row i^{th} column, $\mathcal{A}_{ji}$. Hence, $\mathcal{A}$ is a symmetric matrix. If two nodes, i and j are adjacent, then we have that $\mathcal{A}_{ij} = \mathcal{A}_{ji} = 1$. All other indices are 0. That is for each pair of nodes, we have:

$$\mathcal{A}_{ij} = \mathcal{A}_{ji} = \begin{cases} 1; & i \text{ and } j \text{ are adjacent} \\ 0; & \text{otherwise.} \end{cases} \tag{1}$$

A matrix related to the adjacency matrix is the weighted adjacency matrix, $\mathcal{W}$. In a weighted adjacency matrix, connectivity is defined by the *cost* it takes to travel between 2 connected nodes. These costs replace the 1's in the matrix and weights can be thought of as distances, prices, and lengths of time. We will utilize a distance weighted adjacency matrix in our model.

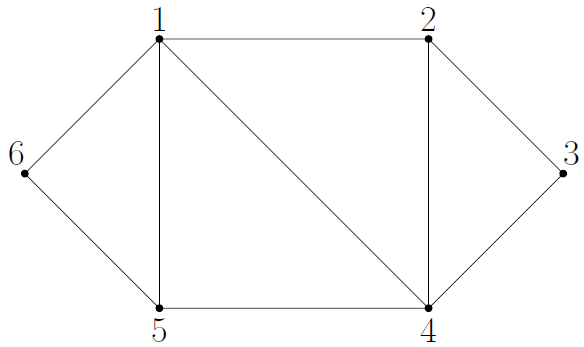
In addition to the adjacency matrix an important matrix we define for a graph G is its *degree matrix*, denoted $\mathcal{D}$. This is simply an $m \times m$ diagonal matrix whose diagonal is the degree of the nodes of G, where the degree of a node is defined to be the number of nodes adjacent to it. That is, for a node i, the i^{th} row i^{th}column of $\mathcal{D}$, denoted $\mathcal{D}_{ii}$, is the degree of the node i. The matrix $\mathcal{D}$ is defined explicitly by:

$$\mathcal{D}_{ij} = \begin{cases} \text{degree of a node } i; & \text{if } i = j \\ 0; & \text{otherwise.} \end{cases} \tag{2}$$

Next, we define the graph's *discrete Laplacian matrix*, denoted with $\mathcal{L}$, to be the degree matrix subtracted by the adjacency matrix or,

$$\mathcal{L} = \mathcal{D} - \mathcal{A}. \tag{3}$$

Finally, we define the *random walk normalized Laplacian* denoted as $\mathcal{L}^{rw}$ (Riascos & Mateos, 2014). This matrix is a modified Laplacian matrix where the elements of each row are divided by the degree of the node corresponding to said row. Figure (1) demonstrates these matrices in use.



$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \quad \begin{bmatrix} 4 & -1 & 0 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & -1 & 4 & -1 & 0 \\ -1 & 0 & 0 & -1 & 3 & -1 \\ -1 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

Adjacency Matrix **Degree Matrix** **Laplacian Matrix**

$$\begin{bmatrix} 1 & -1/4 & 0 & -1/4 & -1/4 & -1/4 \\ -1/3 & 1 & -1/3 & -1/3 & 0 & 0 \\ 0 & -1/2 & 1 & -1/2 & 0 & 0 \\ -1/4 & -1/4 & -1/4 & 1 & -1/4 & 0 \\ -1/3 & 0 & 0 & -1/3 & 1 & -1/3 \\ -1/2 & 0 & 0 & 0 & -1/2 & 1 \end{bmatrix}$$

Random Walk Normalized Laplacian

Figure 1: An example of a network with 6 nodes and its adjacency, degree, and Laplacian matrices.

In practice, one uses the adjacency matrix to determine the neighbouring nodes of a given point. For example, in Figure (1) we see that row 2 of the adjacency matrix has three 1's which correspond to its neighbours: 1, 3, 4. Once can find a cost radius by a similar means with the weighted adjacency matrix. In our model, we will apply the use of the weighted adjacency matrix, which we will refer to as the distance matrix, to define a search radius for our offenders.

We note that the distance metric applied in the model is the *taxicab metric* or the *Manhattan distance metric*. This is because streets do not follow a uniform geometry in general and as such, we must consider distances along streets. Additionally, we will denote an arbitrary point on our network, a site, by the letter s.

Using this distance matrix, network theorists can follow adjacencies to determine distances between intersections which do not have a direct road connecting them. The value in this is one can then determine distances between every intersection on the street network. In reality, individuals cannot take straight line paths to other areas and their mobility and distance travelled is constrained by the street network. On a street network, a distance radius from an intersection, is everywhere an individual can travel to from the given intersection within the defined radius. Under the setting of network theory, distances are measured by what is known as block distances describing the distances for paths between points.

2.2 Random Movement

The driving mechanisms behind the diffusion papers' models are some kind of a random walkers, our criminal agents. The models utilize three main types of random movement *Biased Random Walks* (BRW), *Lévy Flights* (LF) and *Truncated Lévy Walks* (TLW). For our framework, we utilize the latter of these three, the TLW. Movement determined by these three types of random motion are dictated by some sort of weight or attractiveness that varies from site to site with time. Hence, we denote this attractiveness at a given site *s* at time *t* to be $A_s(t)$. Therefore, a network with nodes *1 … m* has an attractiveness field described by the vector below:

$$A(t) = [A_1(t) \dots A_m(t)]^T. \tag{4}$$

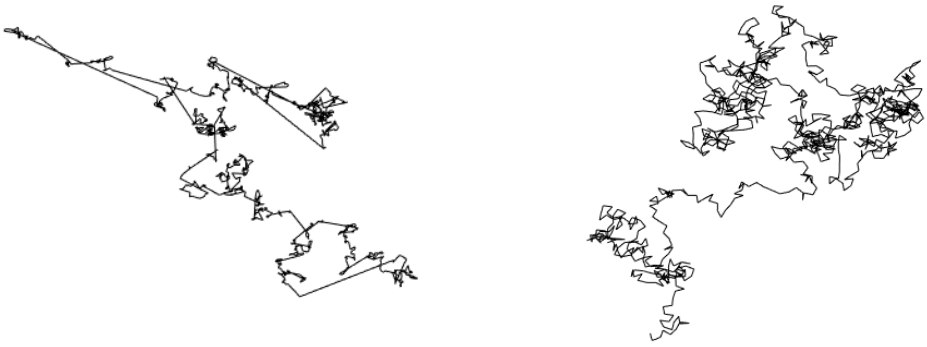
As mentioned, motion defined TLWs is a stochastic process. In a TLW, a random walker can take long steps based on a Truncated Lévy distribution (Pan et al., 2018). The probability distribution is defined by the probability of a random walker jumping from site *a* to *b* as

$$q_{a\to b}(t) = \frac{w_{a\to b}(t)}{\sum_{s=\text{sites not equal to } a} w_{a\to s}(t)}, \tag{5}$$

where we define $w_{a\to b}$ to be

$$w_{a\to b}(t) = \begin{cases} \dfrac{A_b(t)}{l^{\mu} d(a,b)^{\mu}}; & 1 \le d(a,b) \le L \\ 0; & \text{otherwise.} \end{cases} \tag{6}$$

Here, l is a *fineness* or spacing between points in space, μ is a constant between the numbers 1 and 3 serving as the exponent of the underlying power law of the Lévy distribution and *d(a,b)* is the distance between sites *a* and *b* under the *Manhattan distance metric* bounded by a constant *L*, some maximum distance (Pan et al., 2018). TLWs are similar to Lévy Flights which allow for infinite jumps, the case for when $L \to \infty$. We argue that due to the constraints of a city, a random walker cannot take these leaps which was applied in a model by Chaturapruek et al. (2013). As such, we base our model on TLWs similar to a framework by Pan et al. (2018). Figure (2) presents examples of a TLW and a BRW, a type of random motion where step sizes are fixed.



(a) (b)

Figure 2: (a) Truncated Lévy Walk (b) Biased Random Walk

2.3 Diffusion on Networks

Diffusion on networks follows conventional descriptions of diffusion on lattices. Suppose we have some fluid and for any site *s* on our network, we define $\mathcal{F}_s(t)$ to be the amount of that fluid site *s* has at time *t*. Therefore, in a system with nodes 1 ... *m*, the state of fluid allocation throughout the *m* nodes at time *t* is described by the vector below:

$$\mathcal{F}(t) = [\mathcal{F}_1(t) \dots \mathcal{F}_m(t)]^T. \tag{7}$$

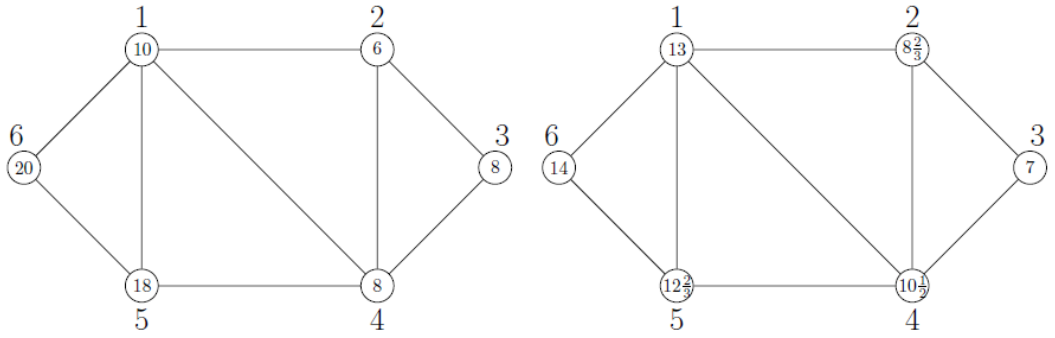
Suppose we pick a site *p* that has *n*-many neighbours. After a time increment of δt, the substance spreads out evenly to all of *p*'s neighbours. That is, $\mathcal{F}_p(t)$ loses $\mathcal{F}_p(t)$ units of fluid *f* while each of its neighbours receives $\mathcal{F}_p(t)/n$ units after a δt timestep. Moreover, $\mathcal{F}_p(t)$ increases by what its neighbours contributed after the given time step. This is done with ease by using the $\mathcal{L}^{rw}$, the *random walk normalized Laplacian* using the update formula (Riascos & Mateos, 2014)

$$\mathcal{F}_p(t + \delta t) = \mathcal{F}_p(t) - \mathcal{L}_p^{rw} \cdot \mathcal{F}(t) \tag{8}$$

where we see $\mathcal{F}_p(t)$ subtracted by the dot product of $\mathcal{L}_p^{rw}$, the p^{th} row of the $\mathcal{L}^{rw}$ matrix, and $\mathcal{F}(t)$, the state of the substance allocation throughout the entire network at time *t*. Furthermore, this generalizes to an operation which updates the entire system using the update formula below:

$$\mathcal{F}(t + \delta t) = \mathcal{F}(t) - \mathcal{L}^{rw}\mathcal{F}(t). \tag{9}$$

We illustrate an example of this in Figure (3), our graph in Figure (1) with an arbitrary initial state of substance allocation.



(a) Time: t (b) Time: $t + \delta t$

$$\mathcal{F}(t+\delta t) = \mathcal{F}(t) - L^{rw}\mathcal{F}(t) = \begin{bmatrix} 10 \\ 6 \\ 8 \\ 8 \\ 18 \\ 20 \end{bmatrix} - \begin{bmatrix} 1 & -1/4 & 0 & -1/4 & -1/4 & -1/4 \\ -1/3 & 1 & -1/3 & -1/3 & 0 & 0 \\ 0 & -1/2 & 1 & -1/2 & 0 & 0 \\ -1/4 & -1/4 & -1/4 & 1 & -1/4 & 0 \\ -1/3 & 0 & 0 & -1/3 & 1 & -1/3 \\ -1/2 & 0 & 0 & 0 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 6 \\ 8 \\ 8 \\ 18 \\ 20 \end{bmatrix} = \begin{bmatrix} 13 \\ 8^2/_3 \\ 7 \\ 10^1/_2 \\ 12^2/_3 \\ 14 \end{bmatrix}$$

Figure 3: Diffusion on a graph with 6 nodes
(a) The graph with arbitrary initial conditions (b) The graph after one time step δt

3. Methodology

We now proceed with an explanation of our methodology. The underlying system for each model has two main components -- stationary crime sites and criminal agents which move from site to site. This system evolves over time with time increments of $\delta t > 0$. For us, each site is a discrete

point on our road network. Hence, each site is represented as a node. To accurately portray offender movement, the road network must be subdivided into street segments of equal lengths 10m. This value 10m will be used in the TLW of criminal agents in our model. We will assume that after subdivision, the network will have m-many sites.

As mentioned, these criminal agents are located at each site. We shall denote the number of offenders at site s at time t as $n_s(t)$. Therefore, the number of offenders throughout the whole system at time t is described by the vector below:

$$n(t) = [n_1(t) \dots n_m(t)]^T. \tag{10}$$

In addition to $n(t)$, the network has an underlying "attractiveness field" defined in Equation (4). As its name suggests, the site's "attractiveness" is an offender's beliefs about the vulnerability and intrinsic value of criminal opportunity at the site at a given time. We assume that this attractiveness consists of the sum of a static background attractiveness A_s^0 and a dynamic attractiveness $B_s(t)$ forming:

$$A_s(t) = A_s^0 + B_s(t). \tag{11}$$

Here, the dynamic attractiveness term $B_s(t)$ represents the component associated with repeat victimization and the broken window effect. Therefore, the system's attractiveness *A(t)*, static attractiveness A_s^0, and dynamic attractiveness $B_s(t)$, can be represented by the following vectors in Equations (4), (12), and (13):

$$A^0 = [A_1^0 \dots A_m^0]^T \tag{12}$$

$$B(t) = [B_1(t) \dots B_m(t)]^T. \tag{13}$$

We note that A^0 can be uniform or can vary across the network.

Hence, each site *s* has an associated vector, a qualitative description, representing the "attractiveness" at time t and the number of criminal agents. This vector is defined to be the pair $\big(A_s(t), n_s(t)\big)$. The whole system then, can be defined similarly by Equations (4) and (10). This system evolves from starting state with an initial distribution of criminal agents and an attractiveness field over the road network. After a time increment, the system will evolve with the following four step process.

Step 1. Generate criminals on the network based off of a Normal Distribution for expected number of crimes for a given time increment. For example, a time step could represent a month and so, the expected number of crimes would have to reflect expected criminal events in the time span of a month.

Step 2. Allow each criminal to search for crimes moving using TLWs. At every site they visit, each criminal decides to whether or not they will commit a crime with the probability

$$p_s(t) = 1 - e^{-A_s(t)\delta t}. \tag{14}$$

If they choose not to commit a crime, they continue searching upwards of a maximum number of searches. If the offender reaches this limit, we assume they have been stuck searching in a given

area and we force said criminal to commit a crime at their last visited site. This limit reduces computational load and accounts for finite time.

Step 3. Once all agents have committed their offences or have run out of steps, they will be immediately removed from the system. This can be thought of the agent fleeing the site of the crime. For each site *s*, we define $E_s(t)$ to be the number of crimes committed there at time *t*. If a crime is committed at site *s* at time *t*, we increment $E_s(t)$ by 1. So, the vector

$$E(t) = [E_1(t) \dots E_m(t)]^T \tag{15}$$

determines the number of crimes committed throughout the system at time *t* and is determined by the events in Step 2.

Step 4. Finally, after every criminal agent has committed a crime and the vector $E(t)$ is updated, the dynamic term is updated by the following update rule (Short et al., 2008)

$$B(t + \delta t) = [B(t) - \eta \mathcal{L}^{rw} B(t)](1 - \omega \delta t) + \theta E(t) \tag{16}$$

where $\eta, \omega,$ and θ are absolute constants each serving different purposes. The constant $\eta \in (0,1)$ represents the strength of the near-repeat victimization effect, ω is the decay rate, and θ is the amount by which local attractiveness at each site gets increased by given the occurrence of a criminal event. Here, the $B(t) - \eta \mathcal{L}^{rw} B(t)$ term accounts for partial diffusion of attractiveness after a δt time step. Additionally, the $(1 - \omega \delta t)$ terms accounts for the decay of perceived attractiveness at each site. Therefore, we find that the update rule diffuses the attractiveness of a given site to its neighbours, decays over time, and gains attractiveness based on near-repeat victimization.

In practice, we advise seeding simulations with empirical data for more realistic results. For example, in our model, we update $E(t)$ using real crime data and using time increments which reflect the expected number of crimes. Tables 1 and 2 outline our fields and parameters respectively while Figure (4) summarizes our four-step.

Fields	
Dynamic Attractiveness	$B(t) = [B_1(t) \dots B_m(t)]^T$
Static Attractiveness	A^0
Attractiveness field	$A(t) = [A_1(t) \dots A_m(t)]^T$ $A_s(t) = A^0 + B_s(t)$
Offenders	$n(t) = [n_1(t) \dots n_m(t)]^T$
Criminal events	$E(t) = [E_1(t) \dots E_m(t)]^T$

Table 1: Fields

Parameters	
Truncation limit	$L > 0$
Fineness	$\ell > 0$
Lévy Distribution Constant	$\mu \in (1,3)$
Discrete time increment	$\delta t > 0$
Diffusivity	$\eta \in (0,1]$
Random-walk normalized Laplacian of network	$\mathcal{L}^{rw}$
Decay rate	$\omega \in (0,1)$
Effect of near-repeat victimization	$\theta > 0$

Table 2: Parameters

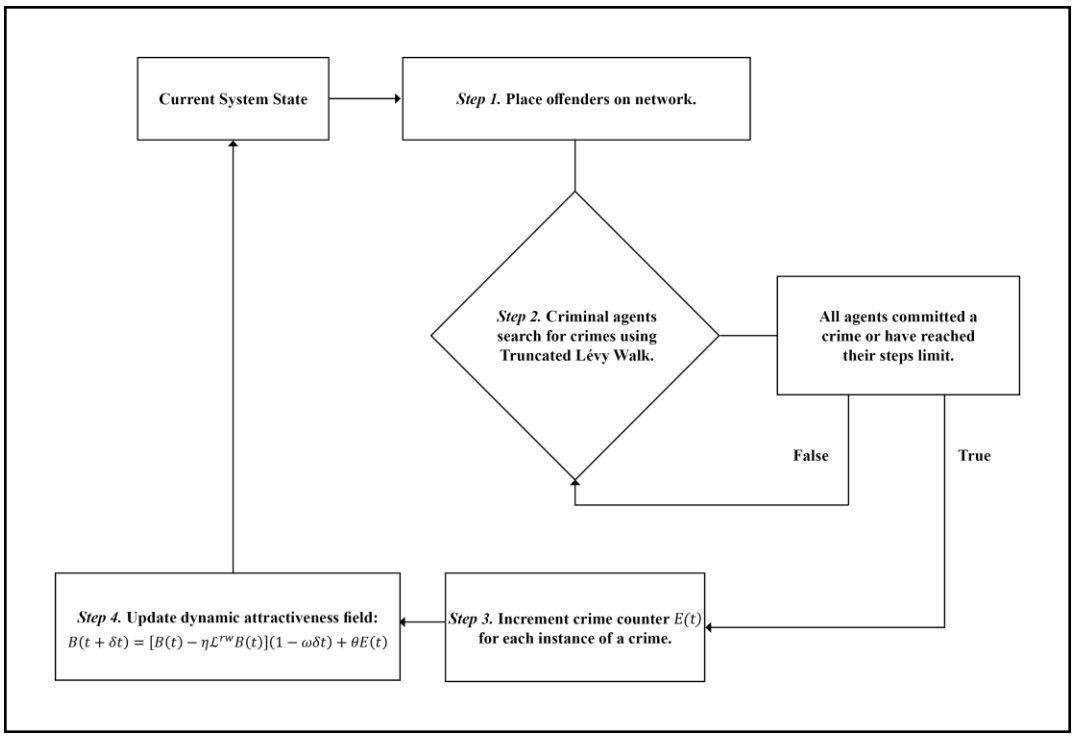


Figure 4: Summary of four step process.

4. Simulations of Crime

4.1 Initialization and Parameterization of the Model

In practice, one can use our model to simulate criminal events which are specific to a given city. To do so, one requires access to data related to the road network and crimes for their city of interest. The available crimes data should be geocoded so that it can be used on the geography of the road network. Our research focused on the city of Vancouver and so, in this technical appendix, the results from this research will be used as an example.

Firstly, the road network of interest must contain sites for the criminal agents to travel to and commit their crimes. In our example, generate sites on the network by subdividing Vancouver's network into 10m segments and crime sites are located at each end of the segments. Once the network data is introduced, one can initialize the attractiveness field for the network.

Initially, the attractiveness field should be non-existent. That is, it is zero throughout the network. Parameterization in Equation (16) will vary, however Table 3 outlines parameters we used in our example. Moreover, choice of δt is dependent on crime data. That is, if the crime data is available for every given day, then $\delta t = 1$ should be used to represent 1 day. Similarly, $\delta t = 30$ for a month should be used to represent 30 days.

Suggested Parameters	
θ	0.1
ω	1/40
η	0.01, 0.1, or 1

Table 3: Suggested parameters for Equation (16)

The vector $E(t)$ is also dependent on the available crime data. Crime data must be organized by time and must be geographically located on the closest available site on the network. Each increment in time of this data will be equal to δt and $E(t)$ will be updated by the crime data. No criminal agents should be put on this network during this initialization state.

The result of this initialization and parameterization will be unique Attractiveness fields which we refer to as seeds. These seeds will be what the simulations of crime with our criminal agents will be run on. We present seeds used in our research in Figure (5).

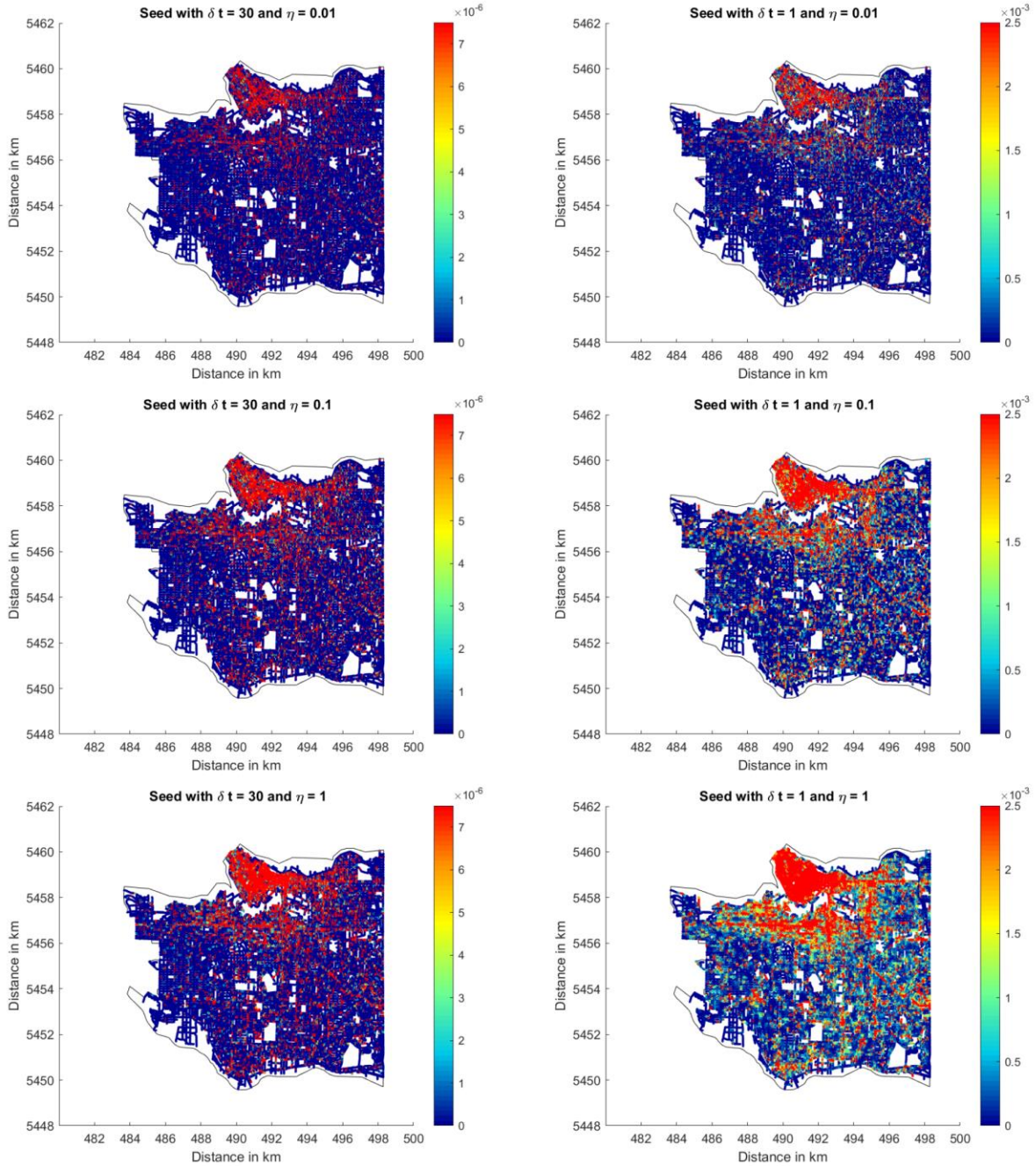


Figure 5: Seeds of crime concentrations used in simulations.

4.2 Error analysis

Of course, different parameters will lead to different results in simulations. To determine optimal parameters, one should perform an error analysis to determine the accuracy of a seed. To measure the magnitude of this error we perform a Mean-Squared Error (MSE) analysis. A smaller MSE implies greater accuracy in a given simulation, in contrast to a simulation with a higher MSE.

However, due to the randomness of the criminal agents, we advise performing a Monte Carlo Algorithm on each seed with at least 200 trials and averaging the MSE over all trials. Trials should be grouped by different seeds to test for performance changes when changing parameters. By performing the Monte Carlo Algorithm on these different groupings, one accounts for all the randomness between trials for simulations to derive a more accurate error term.

In our research, we tested the effects of diffusiveness, truncation limits, and the scale of different time steps on the error of our simulations. The error of our simulations was determined by comparing it directly to data for crime in Vancouver in June 2019 (see Figure (6)). Each of these tests had the same θ and ω values outlined Table (3) to maintain consistency between tests. The values of the diffusiveness η, truncation limit L, and time steps δt are outlined in Table (4). These tests resulted in the error graph found in Figure (7) with the most accurate result corresponding to parameters $\eta = 0.1$, $L = 2000$, and $\delta t = 1$.

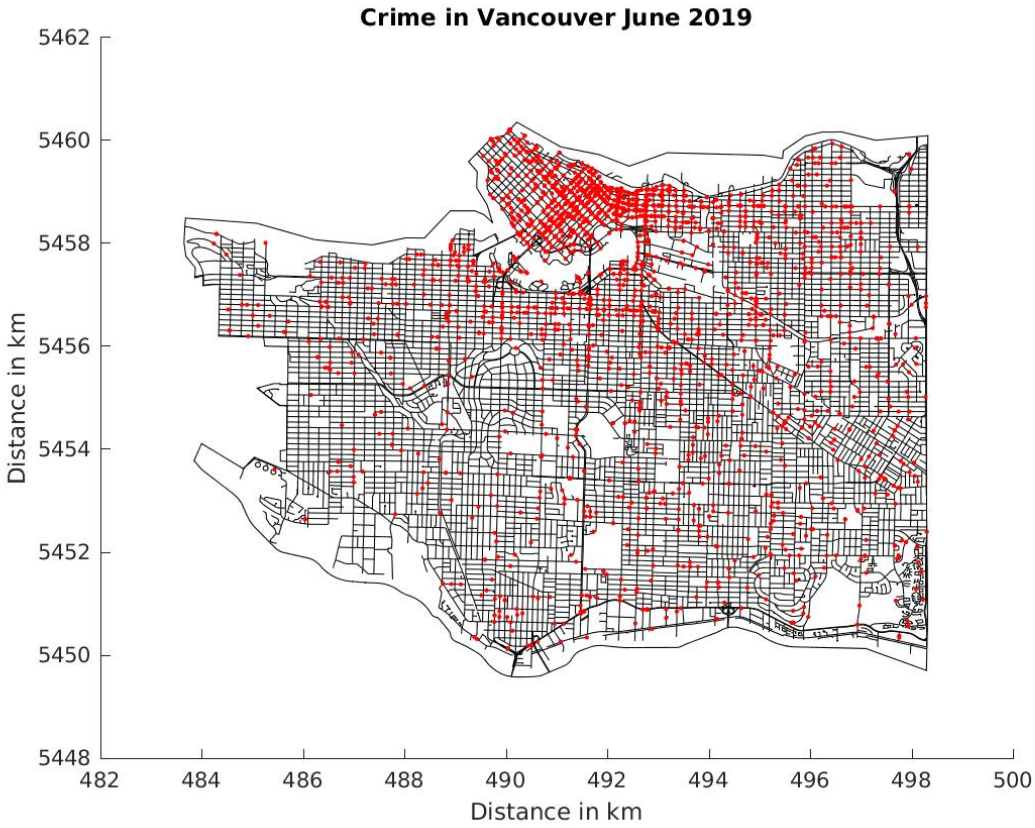


Figure 6: Most accurate trial from simulations.

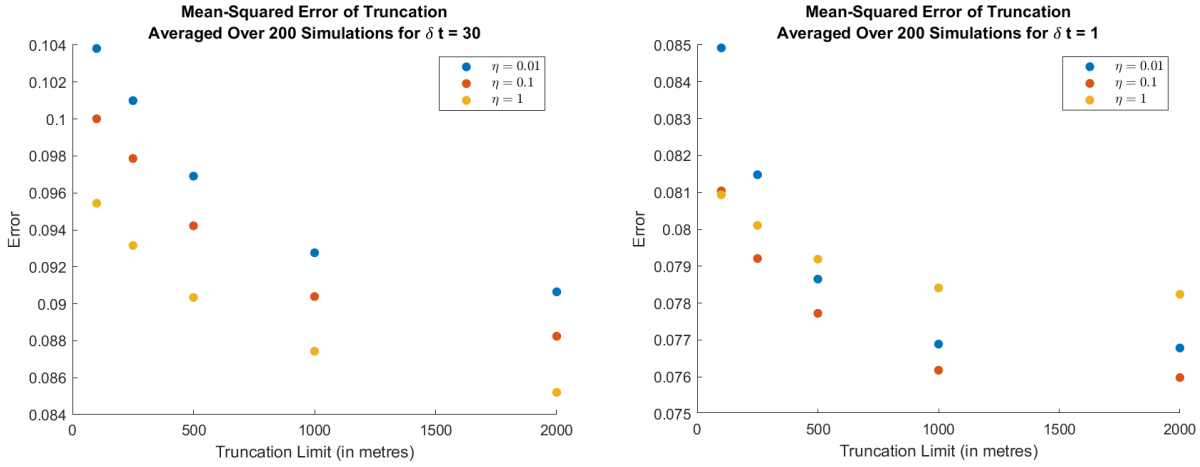
Variable Parameters	
η	0.01, 0.1, 1
L	100, 250, 500, 1000, 2000
δt	1, 30

Table 4: Values of parameters which varied between tests

5. Discussion of Results

We now present results from computer simulations of our framework. These results illustrate the effects of different parameterizations in the model and the magnitude of errors when compared to crimes in June 2019 (See Figure (6)). To determine the magnitude of these errors, we perform what is called a Mean-Squared Error (MSE) Analysis across the network. A lower MSE corresponds to a more accurate parameterization while a higher MSE corresponds to a less accurate one.

From the MSE analysis in Figure (7), we find that the MSE corresponding to a discrete time increment of $\delta t = 30$ decreases as we increase the Truncation limit L. With the maximum Truncation Limit in this discrete time increment, we see increased accuracy with high values of diffusivity as well. In fact, the most accurate simulations corresponded to trials which ran with a Truncation limit of $L = 2000$ and diffusivity $\eta = 1$. As $\eta = 1$ is the highest possible value for diffusivity, it appears that this would be the optimal value for $\delta t = 30$.



(a) (b)

Figure 7: Mean-Squared Error of Truncation for different values of δt.
(a) $\delta t = 30$ (b) $\delta t = 1$.

Similarly, for $\delta t = 1$, we find that the MSE decreases as we increase the Truncation limit L as well. However, we find that the results of diffusiveness are not similar with this discrete time increment. In fact, greater diffusiveness, $\eta = 1$, lead to greater errors than in lower levels of diffusiveness, $\eta = 0.01$ and 0.1, as Truncation limits increased. With Truncation limits smaller than 500, the results from these trials had errors less predictable than that of $\delta t = 30$. Despite this, for limits greater than 500 there is a more predictable trend which is in line with the errors we saw for $\delta t = 30$. We also see the errors leveling out at $L = 2000$.

In light of this however, each simulation run with the smaller time increment yielded errors smaller than all results obtained from the discrete time increment of 30. The most accurate of these simulations corresponds to the values of $L = 2000$ and $\eta = 0.1$. In Figure (8), we present the trial with the smallest MSE for $\delta t = 1$. Comparing Figures (6) and (8), one can see apparent visual similarities between our simulation and the Vancouver crime in June 2019.

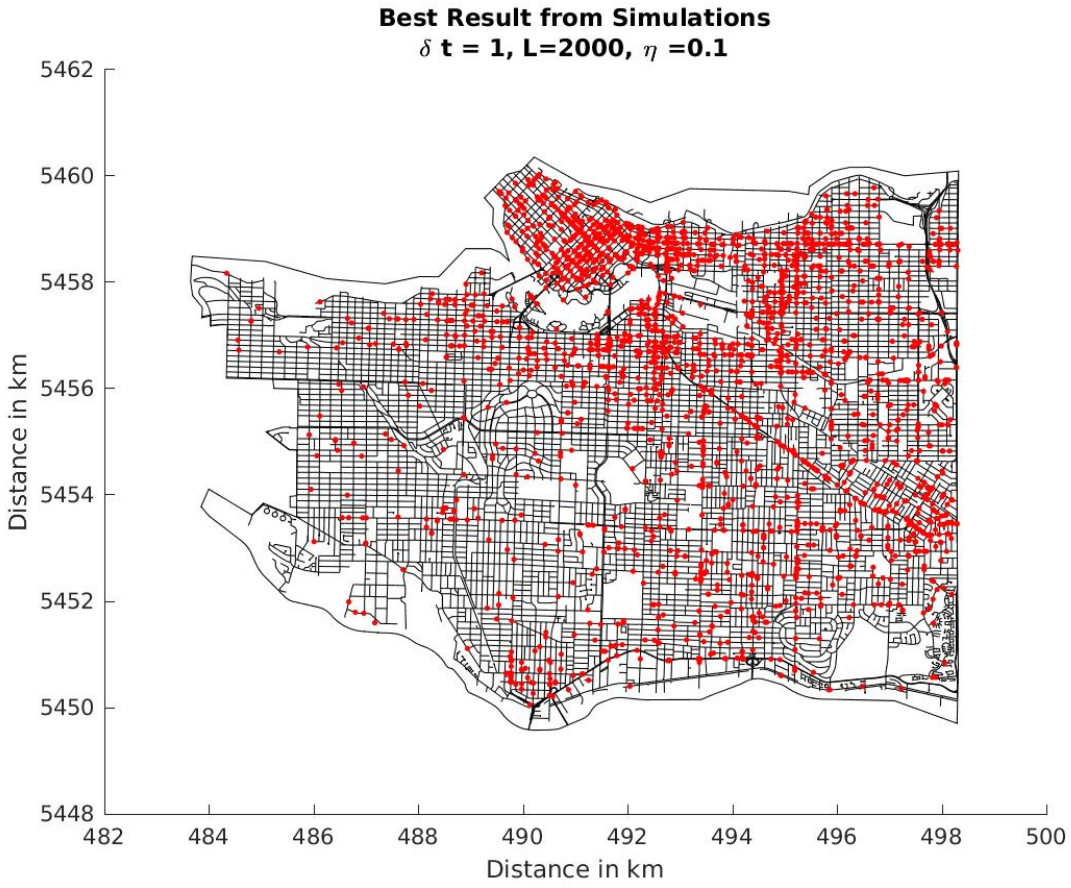


Figure 8: Most accurate trial from simulations.

This increase in accuracy could be justified by the short amount of time between crimes corresponding to the effects of near-repeat victimization. That is, the effects of near-repeat victimization are most prominent within a short time window after a criminal event. Clearly, 1 day is a much smaller time window than 30 days. We argue that the accuracy of smaller diffusiveness, $\eta = 0.1$, in the smaller time increments, $\delta t = 1$, is in line with the value for $\eta = 1$ for the time increment of 30. That is, they represent similar amounts of spreading within the same time period with differences in scale and magnitude.

6. Bibliography

Chaturapruek, S., Breslau, J., Yazdi, D., Kolokolnikov, T., & McCalla, S. G. (2013). Crime Modeling with Lévy Flights. *{SIAM} Journal on Applied Mathematics*, *73*(4), 1703–1720.

Pan, C., Li, B., Wang, C., Zhang, Y., Geldner, N., Wang, L., & Bertozzi, A. L. (2018). Crime modeling with truncated Lévy flights for residential burglary models. *Mathematical Models and Methods in Applied Sciences*, *28*(09), 1857–1880.

Riascos, A. P., & Mateos, J. L. (2014). Fractional dynamics on networks: Emergence of anomalous diffusion and Lévy flights. *Physical Review E*, *90*(3).

Short, M. B., D'Orsogna, M. R., Pasour, V. B., Tita, G. E., Brantingham, P. J., Bertozzi, A. L., & Chayes, L. B. (2008). a Statistical Model of Criminal Behavior. *Mathematical Models and Methods in Applied Sciences*, *18*(supp01), 1249–1267. https://doi.org/10.1142/S0218202508003029

West, D. B. (2000). *Introduction to Graph Theory* (2nd ed.). Pearson.