Crime Hot Spot Forecasting: A Recurrent Model with Spatial and Temporal Information

Yong Zhuang[1], Matthew Almeida[1], Melissa Morabito[2], and Wei Ding[†1]
[1]*Department of Computer Science*
University of Massachussets Boston
Boston, Massachussets 02125-3393
[2]*School of Criminology and Justice Studies*
University of Massachusetts Lowell
Lowell, Massachussets 01854
Email: [1]{$yong.zhuang001, matthew.almeida001, wei.ding$}@$umb.edu$, [2]$Melissa\_Morabito@uml.edu$

***Abstract*—Crime is a major social problem in the United States, threatening public safety and disrupting the economy. Understanding patterns in criminal activity allows for the prediction of future high-risk crime "hot spots" and enables police precincts to more effectively allocate officers to prevent or respond to incidents. With the ever-increasing ability of states and organizations to collect and store detailed data tracking crime occurrence, a significant amount of data with spatial and temporal information has been collected. How to use the benefit of massive spatial-temporal information to precisely predict the regional crime rates becomes necessary.**

The recurrent neural network model has been widely proven effective for detecting the temporal patterns in a time series. In this study, we propose the Spatio-Temporal neural network (STNN) to precisely forecast crime hot spots with embedding spatial information. We evaluate the model using call-for-service data provided by the Portland, Oregon Police Bureau (PPB) for a 5-year period from March 2012 through the end of December 2016. We show that our STNN model outperforms a number of classical machine learning approaches and some alternative neural network architectures.

1. Introduction

Crime prediction has become an area of significant research interest in recent years. With the ever-increasing ability of states and organizations to collect and store detailed data tracking crime occurrence, the need for methods to effectively analyze that data for patterns in space and time has become more and more important. Extensive criminal justice research [1] suggests that targeting specific high-risk areas called "hot spots" is an effective policing strategy; with an accurate predictive model (Figure 1) able to identify periods of high crime risk in particular areas of a city, police departments would be able to allocate their resources to the hot spots far more efficiently to prevent or quickly respond to criminal activity, incorporating the model into the organizational measures taken by the departments and allowing for the effective deployment of officers to areas of high risk or removal of officers from areas seeing decreasing levels of crime. However, given the volume of data and the high number of variables that crime risk depends on, it is a challenging task to formulate, analyze and predict crime data to develop such a model.

It is useful to imagine each location in a city as having a certain level of crime risk derived from its environmental characteristics - the income level of its residents, its demographic makeup, public transit access, perhaps how well-lit it is at night - and that risk level as determining the likelihood of a crime occurring at that location in a given time interval. Of course, the risk at a location could change over time in both the long- and short-term: day to night, month-to-month, year-to-year. With the ever-increasing ability of states and organizations to collect and store detailed data tracking crime occurrence, a significant amount of data with spatial and temporal information has been collected. The challenge, then, is to develop a method to leverage this large dataset and develop a model that can accurately predict future hot spots.

One of the most common and well studied approaches for time series prediction is the Markov process [2] [3], which formulates the problem as a discrete-time sequence prediction task. It predicts the most likely state to occur on the next step based on a fixed-length sequence of observed states. There is a limit that when the number of states is large, Markov models usually cannot capture long-term dependency based on the history since the overall state-space will grow exponentially in the number of time steps considered. Although variable-order Markov (VOM) [4] [5] models can vary the number of conditioning states based on the specific observed realization and Semi-Markov [6] [7] models the continuous time-interval between two successive states, they still has the same state-space explosion issue when the order grows.

[†]*Corresponding author.*

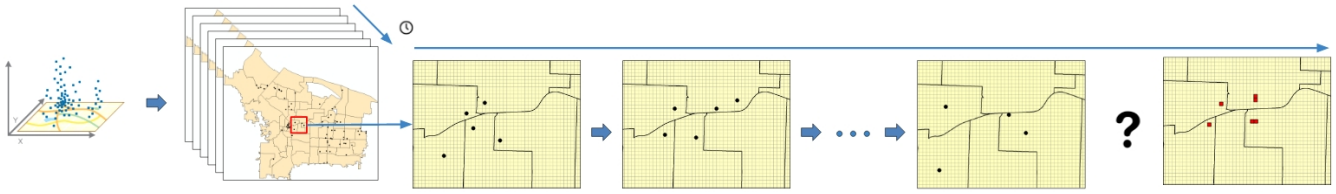

Figure 1. Given the locations and time of historical crimes, can we predict the hot spots in the near future?

Theoretically, recurrent neural networks (RNN) [8], which have been successfully employed for word embedding [11] [21] and formal languages generation [10], are capable of capturing long-distance dependencies. However, due to the gradient vanishing problem, the gradient (error signal) decreases exponentially while the front layers train very slowly, the effect is not ideal. Long short-term memory (LSTM) [9] [20] networks are variants of RNNs which are designed to cope with the gradient vanishing problem. They are well-suited to learn from experience to predict time series when there are time lags of unknown size and bound between important events. However, for a space-time series, which is a sequence taken at a successive equally spaced planes in time and each plane represents objects defined in a geometric space, how can we jointly detect the spatio-temporal patterns from the observations? In this paper, we use the historical data to build a deep recurrent predictive model (STNN) that can detect the spatio-temporal patterns and predict the crime hot spots in the near future by using a deep recurrent neural network architecture. More specifically, our work makes the following contributions:

• We model the task of crime hot spot forecasting as a classification problem on the level of individual geographical areas, using official historical crime data for training and testing.

• We detect a set of potential hot regions that are associated with the patterns of the hot spot mobility, which are helpful to remove noise, reduce the original cellular data to a more manageable size, and identify the potential cells contributing most to predict the hot spots in the near future.

• We model the spatio-temporal influence in a recurrent architecture which jointly detects the spatial and temporal patterns on the individual geographical areas by learning a general representation of the nonlinear dependency over the history.

• We embed the historical spatial influence into a vector state and use it for predicting the cellular crime risk in the near future.

The rest of the paper is organized as follows: We analyze some other approaches to this and similar problems in Section 2, Related Work. We give background information on the crime data and propose the STNN model in Section 3, and discuss the problem formulation in depth in Section 4. We describe our experiments and present our results in Section 5, and give our conclusions in Section 6.

2. Related Work

In this section, we review several types of models for crime hot spot forecasting including historical- influence-based methods, neighborhood-based methods, recurrent neural networks, and spatio-temporal-influence-based methods.

Historical records are the main basis for crime prediction. For example, E. Cesario et al. [17] uses autoregressive models to predict crime in a selected area of Chicago. The authors analyze the number of crimes over time and illustrate the components of the observed number of crimes over time curve, breaking it into trend, seasonal, and random signals to give valuable perspective on the factors that crime risk depends on. However, this analysis holds location constant (that being the designated area of Chicago - their algorithm depends only on time) and aims to do longer-term forecasts than we do in this paper. They perform one- and two-year ahead forecasts where we aim for week- or month-ahead forecasts.

Neighborhood-based methods are commonly used for crime hot spots forecasting with both temporal and spatial information. For instance, D. Wang et al. [14] uses a mapping algorithm to identify connected clusters of cells that together comprise a hot spot. They incorporate demographic features into the historical spatial crime density data to identify their hot spots, but their method is built to work on a single aggregated time step - to locate hot spots given all the crimes that occurred in Portland in December 2016, for example. This approach is unable to use recent nearby events to modify the expectation of the hot spots in the future.

Recently, recurrent neural networks are widely used for time series analysis. Du, et al., in [15], use a deep recurrent neural network to model the conditional intensity function of a general temporal point process problem and test their model against several other model types and synthetic and real-world datasets. In their work, they track specific entities (taxicabs, for example) over time. But when forecasting crime, the hot spots cannot be followed in the same way - we need to model the dependency of the conditional intensity of a particular cell on not only its own history but those of its neighboring regions.

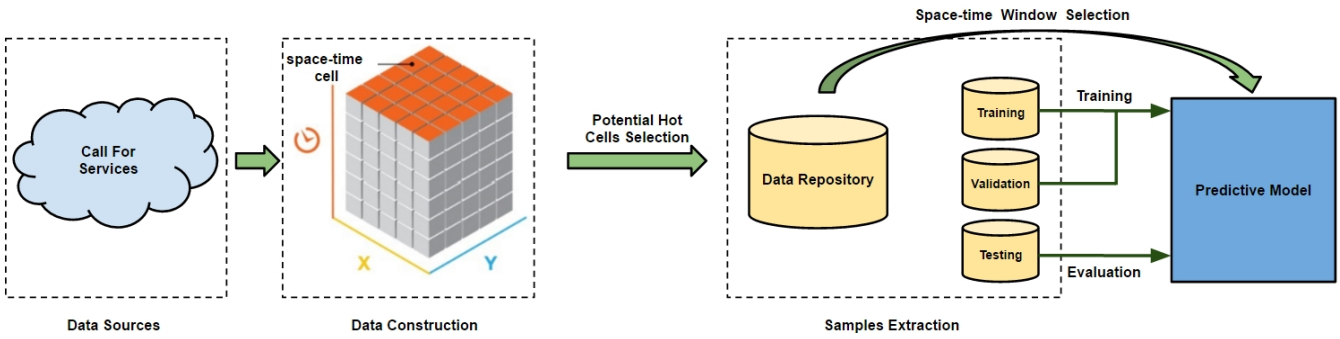

Figure 2. The workflow overview of our study. The predictive model is built through the learning on the potential hot cells with their spatial crime information.

Spatio-temporal-influence-based methods are also popular in space-time series analysis. As in [16], Ratcliffe, et al. do an extensive study on spatial-temporal crime forecasting using data from Philadelphia, PA, and 500ft by 500ft grid cells. They combine long-term demographic-based prediction with short-term event-dependent prediction and incorporate demographic variables at the census-block level. They study four years (2005-2009) of crime data and calculate odds ratios for the increase in conditional intensity exerted by an event on its spatial and temporal neighborhood over that period. A deep neural network may be able to learn more flexible temporal patterns for the spread of an events influence.

In [13], Yu et al. address the crime hot-spot identification problem as working on an entirely burglary dataset, and use boosted trees to attempt to identify grid cells that exhibit correlated behavior - that are commonly hot (or cold) in the same time step. However, their work targets a much larger cell size (800m on a side) than our own (600 ft on a side, over 16x smaller), and we feel that our higher-resolution data is not a good fit to their method.

Based on the discuss above, a major limitation of these existing studies is they can not fully exploit the spatio-temporal information from the space-time series. In this work, we seek to propose a model that can jointly detect the spatio-temporal patterns from the observations.

3. Proposed Model

In This section, we introduce the mobility of the crime hot spots, and then explain how we design our model for crime hot spots forecasting and how to train it.

3.1. Hot Spot Mobility

One of the major questions that presented itself during our analysis was that of the mobility of the hot spots. How likely was it that the areas with the highest crime numbers would be the same month-to-month and year-to-year? If socioeconomic factors remain largely constant over the course of the data, it may be difficult for these patterns to change.

In fact, our analysis of the data-set released by the NIJ for Portland found that the ranking of the hottest cells was very resistant to change. Figure 3 illustrates this. The two horizontal axes show the time index and cell index (one row for each unique cell), so if one was to look down at Figure 3 from above, each point would be a single cell-interval. The vertical axis shows a tiered measure of crime activity and reports whether or not that cell-interval was hot (0) or not (1) by the actual number of crimes in the given threshold.

The cells were indexed by their ranking in the first interval (the cell with the highest number of crimes was given index 0, the next 1, and so on), which is why we see that the predicted hot spots are clustered in those cells with low indices - the cells that were the hottest in the first interval are among the hottest for the entire 5-year period. Cells that were outside the highest 400 cells by number of crimes were almost never hot spots.

Because of this, we focus the training of our model on those areas that are hot spots or border on hot spots.

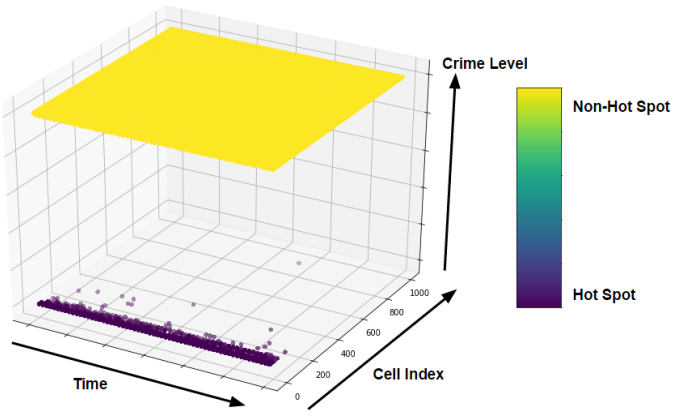

Figure 3. The 3D plot to present the ranking of the hottest cells. The two horizontal axes show the time index and cell index. The vertical axis shows a tiered measure of crime activity and reports whether or not that cell-interval was hot (0) or not(1). It can be seen the ranking was very resistant to change.

3.2. Problem Formulation

Let S be a set of cells, m be the number of crimes, M be a set of m, and $\mathcal{C}$ be the crime level. For each cell s, its location information is associated with its coordinate $\{x_s, y_s\}$, its crime level at a specific time t is denoted as $\mathcal{C}_t^s$, and its historical crime information is given as $M^s = \{m_{t_1}^s, m_{t_2}^s, ...\}$, where $m_{t_i}^s$ presents the number of crimes occurred in the cell s in time interval t_i. And the history of all cells is denoted as $M^S = \{M^{s_1}, M^{s_2}, ...\}$. Then the crime hot spots forecasting problem can be formulated as to predict $\mathcal{C}_t^s$, based on the history M^S.

3.3. Recurrent Neural Networks

Generally, a recurrent neural network is able to combine the input and the hidden state of one step with the inner weight matrices to generate the hidden state on the next step. The hidden state can be computed as follows:

$$h_{t_i}^s = f(a \cdot m_{t_i}^s + b \cdot h_{t_{i-1}}^s) \tag{1}$$

where $h_{t_i}^s$ presents the historical information of a cell s until the time interval t_i, $m_{t_i}^s$ denotes the number of crimes occurred in s at time interval t_i. a and b are the weights. And the activation function $f(x)$ is chosen as a rectifier function as follows:

$$h_{t_i}^s = \max\{a \cdot m_{t_i}^s + b \cdot h_{t_{i-1}}^s, 0\} \tag{2}$$

3.4. Spatio-Temporal Neural Networks

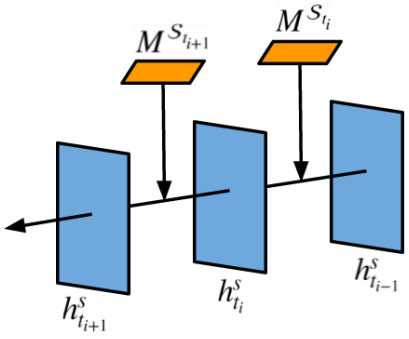

Figure 4. The historical spatial influence

To build a precise space-time series analysis model, both temporal and spatial influence should be considered. In each cell, both its own crime count and that of the cells around it are essential factors for predicting the crime level of the cell, so it is necessary to include all of that information in our model. Therefore the historical spacial influence can be considered as the accumulation of the short-term influences of recent, nearby events. Since influence of a certain event will decrease with the extension of time and space, the historical spacial influence at a cell s can be limited to only those events that have influence that can "reach" that cell, those that are in a particular space-time window $\mathcal{S}_t \subset S$, where $\mathcal{S}_t = \{[t - \Delta t, t], [x_s - \Delta x, x_s + \Delta x], [y_s - \Delta y, y_s + \Delta y]\}$. As shown in Figure 4, given a cell s, its crime level at a specific time t can be denoted as follows:

$$h_{t_i}^s = f(a \cdot M^{\mathcal{S}_{t_i}} + b \cdot h_{t_{i-1}}^s) \tag{3}$$

Here, $M^{\mathcal{S}_{t_i}}$ computes the total spatial influence at cell s in time interval t_i, and it can be described as:

$$M^{\mathcal{S}_{t_i}} = \bigcup_{x_k, y_k \in \mathcal{S}_{t_i}} m_{t_i}^k \tag{4}$$

Finally, for the cell s, we can use $h_{t_i}^s$ to simulate the historical spatial influence of $M^{\mathcal{S}}$ on s, and use this history information to predict crime level of s on the next time interval t_{i+1}. Given the learned hidden state $h_{t_i}^s$, we model the crime level generation with a multinomial distribution by:

$$P(\mathcal{C}_{t_{i+1}}^s = j \mid h_{t_i}^s) = \frac{\exp\left(\mathbf{W}_j h_{t_i}^s\right)}{\sum_{q=1}^{Q} \exp\left(\mathbf{W}_q h_{t_i}^s\right)} \tag{5}$$

where $P(\mathcal{C}_{t_{i+1}}^s = j \mid h_{t_i}^s)$ means the probability of crime level(in cell s at time interval t_{i+1}) equal to j. In this study, we formulate the crime level as a binary classification ($Q = 2$) where hot =0 or non-hot =1. $\mathbf{W}_i$ are the weights of $\mathcal{C}_i$. The architecture of STNN is presented in Figure 5.

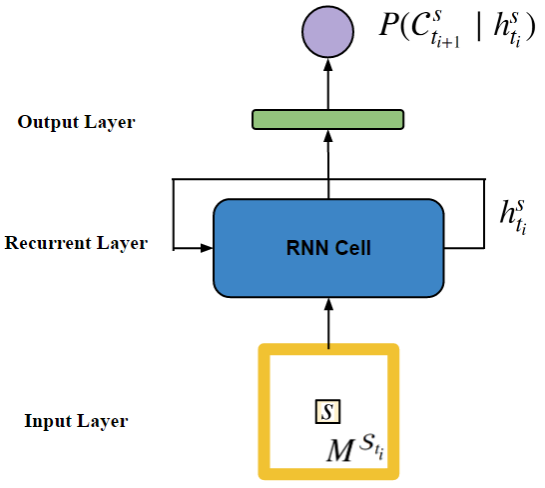

Figure 5. Architect of STNN. For a given sequence M^s, the recurrent layer learns a representation that summaries the nonlinear dependency over the previous events. Based on the learned hidden state h_t^s , it outputs the prediction crime level of cell s on next time step.

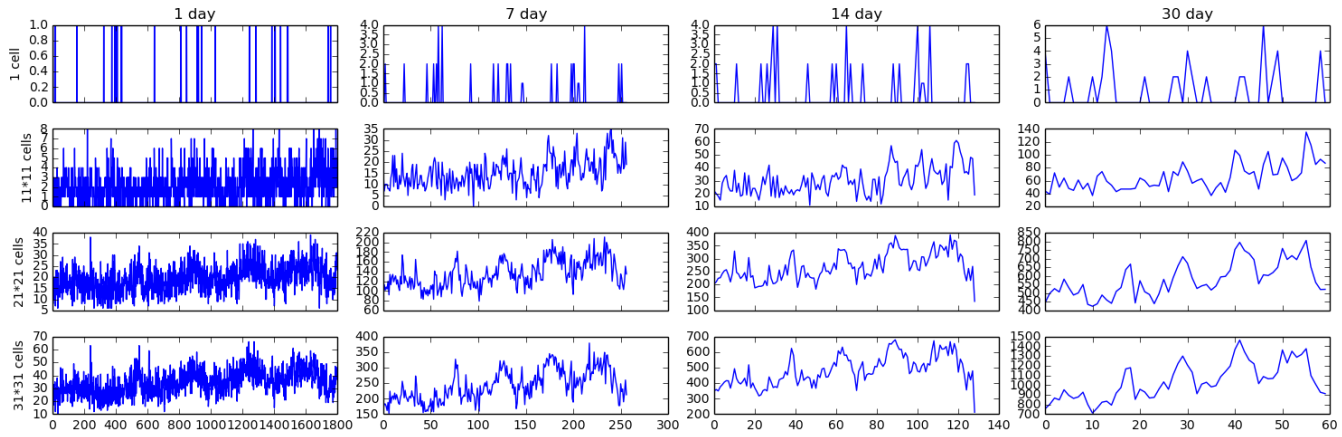

Figure 6. Sequence of the number of crimes in different size regions and time intervals: 1 cell, $11 * 11$ region, $21 * 21$ region, $31 * 31$ region (from top to bottom); 1 day, 7 days, 14 days, 30 days (from left to right).

3.5. Parameter Learning

To optimize the parameters of STNN, we can learn the model by minimizing the joint log-likelihood of $\mathcal{C}$:

$$\mathcal{L} = -\sum_{s} y * log(P(\mathcal{C}^s_{t_{i+1}} = j \mid h^s_{t_i})) \tag{6}$$

We exploit the Back Propagation Through Time (BPTT) [12] for training STNN. Given the number of time steps as n, we unroll our model by n steps, and calculate the gradient at each time step using the back propagation algorithm. We sum up the contributions of each time step to the gradient. Then we employ stochastic gradient descent to estimate the model parameters. This process can be repeated until convergence is achieved.

4. Data Prepossessing

In this section, we introduce the data source used in this study, how we choose the space-time window $\mathcal{S}_t$ over which to calculate $h^s_{t_i}$, and how to pick the potential hot cells.

4.1. Data Source

We use the CFS data provided by the Portland, OregonPolice Bureau (PPB) for a 5-year period from March 2012 through the end of December 2016. In 2016, the National Institute of Justice, the research arm of the U.S. Department of Justice, issued the Real-Time Crime Forecasting Challenge, a competition whereby it could reach out to data scientists from businesses and universities to try to improve spatio-temporal crime forecasting. Call-for-service (CFS) data (samples being requests from residents for police response) was made available for a five-year period from January 2012 through December 2016 from police records in the city of Portland, Oregon, for three CFS types: burglary, street crime, and auto theft.

We extracted from the raw data the longitude and latitude of the locations and their time-stamps. In Figure 7, we converted the geographic map of Portland into a grid of 600×600 cells, which resulted in a grid of size 138×164, and we accumulated the CFS into those cells by 14-day time stamp, as discussed below. This gave us a tensor of size $120 \times 138 \times 164$ describing the entire data-set.

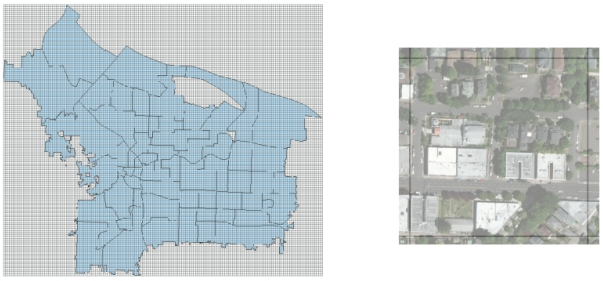

Figure 7. We use a grid size of 138×164, and a cell size of $600 \times 600 ft^2$. The above satellite image gives an example of the cell size for reference.

4.2. Space-time Window Selection

Determining a reasonable space-time window $\mathcal{S}_t$ over which to calculate $h^s_{t_i}$, the overall historical influence on a cell, is very important for model performance. While aggregating the events over a small time-step (say, a single day) allows us to show more detail and variation in the temporal pattern, using such a small interval introduces a large amount of noise in the number of calls for service time-step to time-step. In Figure 6, we compare choices of time-steps by looking at a representative example cell, and vary the spatial window that we examine the time-steps through. The first row considers just crimes in that cell, the next in a 11×11 window, then 21×21, then 31×31. The last three rows show a total number of crimes in the neighborhood for purposes of examining the time windows; when actually training the model, the network input contains the actual crime count for each cell in the neighborhood.

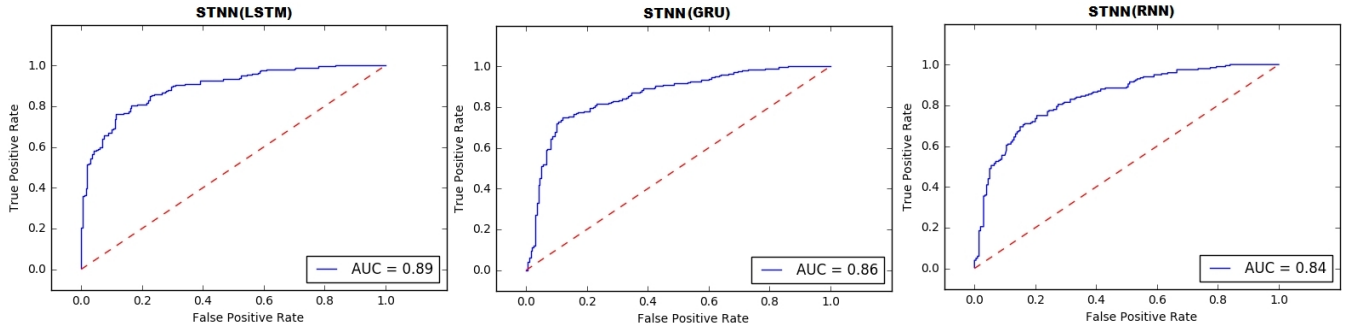

Figure 8. AUC Performance of STNN on three options for RNN architecture.

Increasing the spatial region we look at has a smoothing effect on the number of crimes, as does increasing the size of the time-step. For a single day and single week, the plots are noisy. Using monthly crime counts has a nice smoothing effect, but removes too much temporal information and reduces the overall number of training samples by a significant amount. In this study, we elected to use the space-time window S_t with $Range_t = 14, Range_x = 21, Range_y = 21$.

4.3. Potential Hot Cells Selection

As mentioned in section 3.1 on hot spot mobility, we find that the overall mobility of hot spots is low, and that there are many cells that will never be hot spots. Training the model on temporal sequences from all of the cells in the grid will serve to unbalance the training set and likely bias the model, so we filter the dataset to those cells that are potentially hot spots, like using points-of-interest in taxi destination prediction - locations that are never stopped at or that are only travelled to a few times are not considered. Our training set is selected from the CFS data such that any cell that is a hot spot in any time-step is included in the analysis, and others are removed. Finally, we identified 348 cells that were hot cells at some time step in the dataset.

4.4. Experimental Sample Extraction

For each potentially hot cell s, we calculated the number of calls for service in s and in each cell within a 21×21 square centered on s, for every two-week period starting at the first two-week time-step, March 1-15, 2012. In total, from March 1 2012 to December 2016, we had 129 two-week time-steps.

The model was trained on sequences of 24 2-week time steps, to predict the CFS calls in the 25^{th} 2-week time step. To get the most possible samples out of the dataset, we would shift the time window by one step and then use the new sequence as another sample. So for a given potential hot cell s, we would first train on time steps 0 to 23 and predict 24, then on 1 to 24 and predict 25, etc, which gave us a total of 105 sequences per potential hot cell and 36,540 overall samples.

Of these, only 2,100 were positive samples - meaning that they were in the top cells in the particular time step that was being predicted. To balance the classes, we randomly selected 2,100 samples from among the negative samples to match the number of positives.

These were then randomly divided into training, validation, and test sets, with proportion 80% training, 10%

TABLE 1. THE PARAMETERS FOR EXPERIMENTS ON STNN.

Subject	Parameter	Value
Network	Learning Rate	0.0005
	Input Neurons	441
	Memory Neurons	20
	Time Steps	24
	Number of Class	2
Training	Positive Samples	1,700
	Negative Samples	1,700
	Training Epochs	300
Validation	Positive Samples	200
	Negative Samples	200
Testing	Positive Samples	200
	Negative Samples	200

TABLE 2. PERFORMANCE COMPARISON ON 3 RNN ARCHITECTURES EVALUATED BY ACCURACY, PRECISION, RECALL,AND F1-SCORE.

Models	Accuracy	Precision	Recall	F1-score
STNN –RNN	0.75	0.865	0.703	0.776
STNN –LSTM	0.815	0.87	0.745	0.801
STNN –GRU	0.815	0.862	0.75	0.802

TABLE 3. PERFORMANCE COMPARISON ON STNN (LSTM) AND 6 BASELINE METHODS EVALUATED BY ACCURACY, PRECISION, RECALL,AND F1-SCORE.

Models	Accuracy	Precision	Recall	F1-score	Parameters		
STNN –LSTM	0.815	0.87	0.745	0.801	Memory Neurons 20	Learning Rate 0.0005	Activation relu
Decision Tree	0.76	0.806	0.685	0.74	Criterion gini	Max Depth 5	
Gaussian Naive Bayes	0.743	0.79	0.66	0.719			
Random Forests	0.7625	0.803	0.695	0.745	Estimators 10	Min Samples Split 2	
K-nearest neighbors	0.6375	0.71	0.62	0.662	K 1	Distance Measure L2	
Logistic Regression	0.75	0.767	0.725	0.746	Training Epochs 300	penalty L2	
Multi-layer Perception	0.7675	0.77	0.766	0.768	Hidden Layer Size (100,50)	Learning Rate 0.001	Activation relu

validation, and 10% test.

5. Experiments and Results

In this section, we conduct experiments to evaluate the effectiveness of STNN. We first compare STNN with 3 state-of-the-art RNN architectures, Then we use STNN with the best setting to compare with several state-of-the-art baseline classification models.

5.1. Comparisons with Three Options for the RNN Architecture

First, we construct 3 state-of-the-art RNN architectures. The parameters for training STNN are illustrated in Table 1:

Recurrent Neural Network (RNN) [8]: This model uses a traditional looping RNN structure, where the network contains a hidden layer that uses the output of the previous time step as an additional input to the analysis of the current time-step. Layers use a tanh activation neuron as the nonlinearity to produce output.

Long Short-Term Memory (LSTM) [18] LSTM is a popular variation on the traditional RNN model. Standard RNN has an issue where the ability of the model to remember important information about a previous time step decays as the model looks at more and more tokens. LSTM addresses this by introducing the input and forget "gates", neurons with weights that allow the network the learn how much it should remember or forget about the history, relative to its input at the current step.

Gated Recurrent Unit (GRU) [19]: GRU is a further modification of the LSTM architecture which combines the forget and input gates of an LSTM network into a single update gate. Often, this leads to a simpler recurrent network with performance very close to that of the more complex LSTM architecture.

We implemented our model using Google's TensorFlow, and evaluated three options for the RNN architecture using several metrics. Accuracy, Precision, Recall, and F1-score are standard metrics for ranking tasks. The larger the value, the better the performance. The performance comparision is illustrated in Table 2. The GRU version and the LSTM version obtain similar performance in an accuracy of 81.5%, precision 86-87%, recall 75%, and F1-score 0.8. And both of them perform significantly better than traditional RNN version. Based on the AUC performance in Figure 8, the GRU version is 2% better than the RNN version. And the LSTM version achieve the best AUC score, which improve 3% on the GRU version.

5.2. Comparisons with Traditional Classification Algorithms

In this subsection, we compare our model with the state-of-the-art classification algorithms: Decision Trees, Gaussian Naive Bayes, Random Forests, K-Nearest Neigh-

bors, Logistic Regression, and Multi-Layer Perceptron. The results are shown in Table 3. It can be seen that our STNN(LSTM) model performed better than each of the traditional machine learning algorithms.

As covered in subsection 4.4, we used a random selection of 2,100 negative samples to balance the 2,100 hot-spot samples in the dataset when evaluating STNN. To produce the results in Table 3, we did a single selection of the negative samples and used that same set for all the algorithms to prevent any one method from possibly performing better or worse based on a favorable or unfavorable selection of negative samples.

Each algorithm was evaluated using ten-fold cross validation on the training set.

6. Conclusion and Future Work

In this paper, we formulate the problem of crime forecasting as a space-time series prediction problem and implement a corresponding deep recurrent neural network with spatial influence embedding to estimate the crime level in the near future and show that it outperforms conventional machine learning algorithms and alternate choices of architecture.

In future work, we plan to incorporate additional features to the STNN model to see if we can improve model performance, ideally to the point where we can run on a shorter time window. Adding the demographic features to the fully connected layers that calculate the deterministic background rate could help that side of the network.

We also designed this model to operate on our entire CFS dataset, and to treat each call for service the same, which is a naive assumption. In [16], Ratcliffe et al. find that different crime types affect the conditional intensity in different degrees in their Philadelphia study. Expecting that this is also the case in Portland, we think that including the CFS type in the network could also improve model performance.

Acknowledgments

Funding for this research was provided in part by the Oracle Education Foundation grant to the College of Science and Mathematics at the University of Massachusetts Boston.

References

[1] Weisburd, David, and Cody W. Telep, *Hot Spots Policing, what we know and what we need to know*, Journal of Contemporary Criminal Justice, Vol 30, 2014, pp. 200-220.

[2] Booth, Taylor L, *Sequential machines and automata theory*, 3rd ed. New York, U.S.: Wiley, 1967.

[3] Meyn, Sean P., and Richard L. Tweedie, *Markov chains and stochastic stability*. Springer Science and Business Media, 2012.

[4] Shmilovici, Armin, and Irad Ben-Gal, *Using a VOM model for reconstructing potential coding regions in EST sequences*. Computational Statistics 22.1, 2007, pp. 49-69.

[5] Begleiter, Ron, Ran El-Yaniv, and Golan Yona, *On prediction using variable order Markov models*. Journal of Artificial Intelligence Research 22, 2004, pp. 385-421.

[6] Barbu, Vlad Stefan, and Nikolaos Limnios, *Semi-Markov chains and hidden semi-Markov models toward applications: their use in reliability and DNA analysis*. JSpringer Science and Business Media, Vol 191, 2009.

[7] Doob, Joseph L., and Joseph L. Doob, *Stochastic processes*. New York, U.S.: Wiley, Vol 7. No. 2, 1953.

[8] Graves, A., Liwicki, M., Fernndez, S., Bertolami, R., Bunke, H., and Schmidhuber, J, *A novel connectionist system for unconstrained handwriting recognition*. IEEE transactions on pattern analysis and machine intelligence, 31(5), 2009, pp. 855-868.

[9] Sak, Haim, Andrew Senior, and Franoise Beaufays, *Long short-term memory recurrent neural network architectures for large scale acoustic modeling*. Fifteenth Annual Conference of the International Speech Communication Association, 2014.

[10] Socher, R., Bauer, J., Manning, C. D., and Ng, A. Y, *Parsing with Compositional Vector Grammars*. ACL (1), 2013.

[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J, *Distributed representations of words and phrases and their compositionality*. Advances in neural information processing systems, 2013.

[12] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams, *Learning representations by back-propagating errors*. Cognitive modeling 5.3, 1988.

[13] Yu, C. H., Ding, W., Chen, P., and Morabito, M, *Crime forecasting using spatio-temporal pattern with ensemble learning*. Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer International Publishing, 2014.

[14] Wang, D., Ding, W., Stepinski, T., Salazar, J., Lo, H., and Morabito, M, *Optimization of criminal hotspots based on underlying crime controlling factors using geospatial discriminative pattern*. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Springer Berlin Heidelberg, 2012.

[15] Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L, *Recurrent marked temporal point processes: Embedding event history to vector*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2016.

[16] Ratcliffe, Jerry H., Ralph B. Taylor, and Amber Perenzin, *Predictive Modeling Combining Short and Long-Term Crime Risk Potential, Final Report*. Report to the US Department of Justice, June 2016, Document No. 249934.

[17] Cesario, Eugenio, Charlie Catlett, and Domenico Talia, *Forecasting Crimes Using Autoregressive Models*. Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech) IEEE 14th Intl C. IEEE, 2016.

[18] Hochreiter, Sepp, and Jrgen Schmidhuber, *Long short-term memory*. Neural computation 9.8, 1997, pp. 1735-1780.

[19] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y, *Empirical evaluation of gated recurrent neural networks on sequence modeling*. arXiv preprint arXiv:1412.3555, 2014.

[20] Wang, T., Chen, P., and Li, B., *Predicting the Quality of Short Narratives from Social Media*. The 26th International Joint Conference on Artificial Intelligence, 2017.

[21] Wang, T., Chen, P., Amaral, K., and Qiang, J *An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification*. arXiv preprint arXiv:1609.03663, 2016.