DeepCrime: Attentive Hierarchical Recurrent Networks for Crime Prediction

Chao Huang[1], Junbo Zhang[2,3,4], Yu Zheng[2,3,4,5], Nitesh V. Chawla[1]
University of Notre Dame, USA[1]
JD Intelligent City Research[2]
JD Urban Computing Business Unit[3]
Southwest Jiaotong University, China[4]
Xidian University, China[5]
{chuang7, nchawla}@nd.edu,{msjunbozhang, msyuzheng}@outlook.com

ABSTRACT

As urban crimes (*e.g.*, burglary and robbery) negatively impact our everyday life and must be addressed in a timely manner, predicting crime occurrences is of great importance for public safety and urban sustainability. However, existing methods do not fully explore dynamic crime patterns as factors underlying crimes may change over time. In this paper, we develop a new crime prediction framework–*DeepCrime*, a deep neural network architecture that uncovers dynamic crime patterns and carefully explores the evolving inter-dependencies between crimes and other ubiquitous data in urban space. Furthermore, our *DeepCrime* framework is capable of automatically capturing the relevance of crime occurrences across different time periods. In particular, our *DeepCrime* framework enables predicting crime occurrences of different categories in each region of a city by i) jointly embedding all spatial, temporal, and categorical signals into hidden representation vectors, and ii) capturing crime dynamics with an attentive hierarchical recurrent network. Extensive experiments on real-world datasets demonstrate the superiority of our framework over many competitive baselines across various settings.

CCS CONCEPTS

• **Information systems → Spatial-temporal systems**; **Data mining**;

KEYWORDS

Crime Prediction; Attention Model; Spatio-Temporal Data Mining; Urban Computing; Deep Learning

ACM Reference Format:
Chao Huang[1], Junbo Zhang[2,3,4], Yu Zheng[2,3,4,5], Nitesh V. Chawla[1]. 2018. DeepCrime: Attentive Hierarchical Recurrent Networks for Crime Prediction. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Turin, Italy.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3269206.3271793

CIKM'18, Oct 22-26 2018, Turin, Italy

ACM ISBN 978-1-4503-6014-2/18/10…$15.00
https://doi.org/10.1145/3269206.3271793

1 INTRODUCTION

Crimes (e.g., robbery, rape and murder) severely threaten public safety and have emerged as one of the most important problems countries face [9]. According to the annual crime report, over half a million children and youth aged 10-24 years suffered nonfatal physical assault injuries which are related to stabbings and gun shots [26], and recorded crimes has increased from 2300 to 3000 for every 100,000 people during the period of 1980 to 2000 [41]. Therefore, to improve citizen's life quality, accurate and reliable prediction of crimes is a necessity for helping governments and police departments to effectively prevent crimes from happening and/or handle them efficiently when they occur [19]. In this paper, we aim to predict crimes in each region of a city *before they happen*.

To tackle the crime prediction problem, most of existing techniques utilize the demographic data (*e.g.*, racial composition of population, population poverty level) [2, 8, 11], which fail to capture the dynamic crime patterns in urban space due to the relatively stability of demographic features. Only a small number of schemes been proposed more recently to study the crime prediction problem by exploring the spatial and temporal patterns of crimes [36, 40]. However, these solutions did not fully solve the crime prediction problem in a dynamic scenario where factors underlying crime occurrences may change over time.

Developing such a crime prediction system, however, requires addressing several important technical challenges:

Temporal Dynamics of Crime Patterns. The factors underlying crime occurrences may change over time. For example, crime causality on weekdays may differ from weekends. Traditional forecasting approaches, such as Auto-Regressive Integrated Moving Average (ARIMA) [16] and Support Vector Regression (SVR) [3], assume a fixed temporal pattern of time series, which may become limited. Furthermore, if only recent data is considered to make predictions and historical instances are down-weighted, a lot of useful information (*e.g.*, long-term effects with temporal dependencies of crimes) will be lost, limiting the already sparse crime data.

Complex Category Dependencies. The dependencies between different categories of crimes can be arbitrary since any pair of crime events could potentially be related for different regions. For example, a robbery occurring yesterday may reduce the probability of a future crime occurrence in the region, due to increased patrol in response to the initial robbery. Hence, it is a significant challenge to generalize the crime prediction framework to handle such complex dependencies among different categories of crimes across different regions over time

Inherent Interrelations with Ubiquitous Data. Various ubiquitous data might provide helpful contextual information for capturing crime patterns. First, anomalies in an urban scenario (*e.g.*, blocked driveway and noise) may be considered to be relevant to the crime occurrences. For instance, the occurrence of an assault is likely to cause traffic congestion due to the temporary traffic control by police. Additionally, the citywide Point-of-Interest (POIs) information can characterize the function of each region in a city. Such information could offer insights to advance the understanding of implicit dependencies between crimes occurring in different geographical regions. It is not a trivial task to incorporate both the static (*e.g.*, POIs) and dynamic (*e.g.*, urban anomalies) ubiquitous data into the solution of crime prediction.

Unknown Temporal Relevance. The relevance of crimes across different time frames is unknown. It is not necessary that a future crime occurrence will be more relevant to a recent crime than one that is further away. For example, a crime occurring tomorrow may be related to one occurred yesterday (short-term influence) or last week (periodic influence). Therefore, it is challenging to determine the importance of crimes from previous time steps in assisting the prediction task.

To address the aforementioned challenges in solving the crime prediction problem, we propose a neural network framework DeepCrime to predict the crime occurrences of different categories in each region of a city. First, we develop a region-category interaction encoder to handle the complex interactions between regions and categories of occurred crimes. Second, we propose a hierarchical recurrent framework to jointly encode the temporal dynamics of crime patterns and capture the inherent interrelations between crimes and other ubiquitous data (*i.e.*, urban anomalies and POIs). Finally, we employ the attention mechanism to capture the unknown temporal relevance and automatically assign the importance weights to the learned hidden states at different time frames.

The main contributions of this work are summarized as follows:

- We develop a category dependency encoder that jointly maps the region and crime into the same latent space with their time-evolving correlations preserved.
- We propose a hierarchical recurrent framework that is capable of capturing the dynamic crime patterns and their inherent interrelationships with other ubiquitous data. Furthermore, an attention mechanism is introduced for learning the importance weights of crime occurrences across time frames for making predictions.
- We perform extensive experiments on real-world datasets collected from NYC. Evaluation results demonstrate that the DeepCrime framework significantly outperforms state-of-the-art baselines in terms of prediction accuracy across various settings.

The remainder of this paper is organized as follows. We first formally define the problem in Section 2. We describe the details of our crime prediction model in Section 3. The evaluation results are presented in Section 4. In Section 5, we discuss the related work. We conclude this paper in Section 6.

2 PRELIMINARIES

2.1 Problem Description

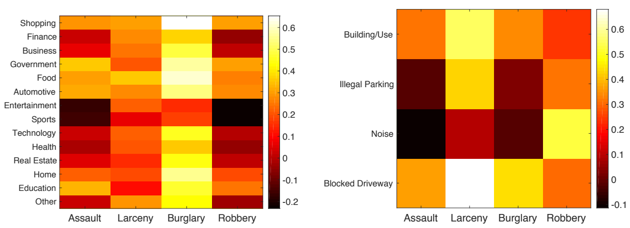
In this section, we begin with some necessary notations and then formally present the problem formulation of crime prediction. Particularly, we consider a set of I geographic regions in a city (*i.e.*, $R = (R_1, ..., R_I)$), a set of J crime categories (*i.e.*, $C = (C_1, ..., C_J)$), and K time slots (*e.g.*, days). We refer to an individual region as $R_i \in R$, an individual crime category as $C_j \in C$, where i, j and k are defined as the index for the region, category and time slot, respectively. We first define the input, extracted from crime data, to our framework.

Definition 1. Crime Matrix $\mathcal{CM}_i$. We define a crime matrix $\mathcal{CM}_i \in \mathbb{R}^{K \times J}$ to represent the crime sequences of all categories in C across K time slots for region R_i. Specifically, in $\mathcal{CM}_i$, we set the element $\mathcal{CM}_i^{k,j} = 1$ if there exists reported crime of category C_j from region R_i in k-th time slot and $\mathcal{CM}_i^{k,j} = 0$ otherwise.

Along with the rapid progress of urbanization, participatory urban sensing-based anomaly reporting systems (*e.g.*, 311 governmental non-emergency services) have been developed to enable pervasive and real-time reporting of anomalies with different categories in a city (*e.g.*, traffic congestion and noise). These reported anomalies widely model urban activities and uncover urban dynamics across different regions, which was considered to be highly related to crimes [40]. Furthermore, the distribution of Point-of-Interests (POIs) in a city characterize the functionality of geographical regions in urban space. These POIs belong to different categories, such as food, shopping and business. A recent study has shown that such POI information is relevant to crime rate analysis [27].

To quantify the relations between crimes and the aforementioned data sources (*i.e.*, POIs and 311 anomalies), we investigate their correlations on the real-world datasets from New York City, *i.e.*, crime data, POI data and 311 urban anomaly data (we present the details of these datasets in Section 4). In particular, we first generate a vector to reflect the frequency of crime occurrence with category C_j for all regions. Each element in this vector is the number of days crimes with category C_j occur. Next, we generate another two vectors with the same length for POIs and anomalies, respectively, as: (i) *POIs*: we generate a vector to represent the density POIs for all regions in a city. Each element in this vector is the number of venues belonging to p-th POI category in region R_i. (ii) *Anomalies*: we generate a vector to indicate the frequency of anomaly occurrences. Each element in this vector is the number of anomalies with d-th category reported from region R_i.

Figure 1(a) and 1(b) show the correlation analysis of different data sources as evaluated by Pearson correlation coefficients [1]. From Figure 1(a), we can observe that most categories of POIs are either positively correlated or negatively correlated with crime occurrences. For example, robberies are more likely to happen in regions with denser POIs of the Entertainment category (*e.g.*, bars and clubs). Similarly, we can observe that the urban anomalies are highly related with crimes from Figure 1(b). Hence, we further utilize the POI and urban anomaly data in our crime prediction framework and give the following definitions which serve as the inputs to our model.



(a) POIs vs. Crimes

(b) Anomalies vs. Crimes

Figure 1: Correlation analysis between crime occurrences and the density of POIs as well as frequencies of urban anomaly occurrences.

Definition 2. Urban Anomaly Matrix $\mathcal{AM}_i$. We define an urban anomaly matrix $\mathcal{AM}_i \in \mathbb{R}^{J \times K}$ to indicate the crime sequences of all categories across K time slots for region R_i. Particularly, in $\mathcal{AM}_i$, each element $\mathcal{AM}_i^{j,k}$ is the number of the j-category crimes that happened at region R_i in k-th time slot.

Definition 3. Point-of-Interests (POI) Matrix PM. We define POI Matrix $PM \in \mathbb{R}^{I \times P}$ to represent the semantic function of each region in a city. Specifically, each element $PM_{i,p}$ represents the number of POIs with p-th category at region R_i.

Crime Prediction. With the aforementioned notations and definitions, the problem of crime prediction is formulated as follows: given the crime matrix $\mathcal{CM}_i$, urban anomaly matrix $\mathcal{AM}_i$ generated from historical K time slots for region R_i and POI matrix PM, the objective of this work is to learn a predictive framework which infers the unknown crime occurrence of each category C_j at each region R_i in h future time slots (*i.e.*, $\mathcal{CM}_i^{j,(K+h)}$).

2.2 Framework Overview

DeepCrime is a multi-layer representation learning framework which solves the crime prediction problem formulated above. We present the model architecture in Figure 2, where the output of one layer serves as the input to the next one. Before presenting DeepCrime, we elaborate the motivations of the model design that attempt to address the challenges identified in Section 1.

- To capture the complex temporal dynamics of crime patterns and its interrelations with ubiquitous data, we develop a hierarchical recurrent framework which captures the complex time-evolving dependencies between the crime occurrences in different time slots. In particular, our framework has a three-level Gated Recurrent Units (GRU) architecture, to encode the temporal dependencies of crime sequence, anomaly sequence and their inter-dependencies, respectively. Additionally, DeepCrime incorporates the POI information as constraints into the region embedding learning process.
- To model the inherent region-category interactions, for each region, we first introduce an input weight vector to distinguish the occurrences of which previous crime categories are more important for future predictions. Then, we concatenate the region and crime embedding vector and feed it into a Multilayer Perceptron (MLP) layer to automatically assign an importance weight to each crime category.
- To address the challenge of unknown temporal relevance between past and future crime occurrences, we design an attention layer that models the importance of crime occurrence in each of past time slots for predicting future crimes.
- Finally, the concatenated hidden state from the attention layer is fed into a Multilayer Perceptron neural architecture to map the learned latent vectors to the predictive probability of crime occurrence of each category in each region of a city.

3 METHODOLOGY

In this section, we present the details of our DeepCrime framework. DeepCrime consists of three major modules: *Region-Category Interaction Encoder*, *Hierarchical Recurrent Framework* and *Attention Mechanism*. We will explain these three modules in detail in the following subsections.

3.1 Category Dependency Encoder

To consider the geographical context of regions, we first incorporate the POI information into the process of generating the region embedding vector. Formally, we define the constraint term as follows:

$$Loss_c = \frac{1}{R} |E_R - PM \cdot W_{POI}| \tag{1}$$

where W_{POI} represents the transition matrix which maps POI matrix PM into the same space as the region embedding vector E_R.

In order to capture the inherent dependencies across categories, we first define the input weight vector μ with a size of J and each element μ_j represents the relevance weight between the j-th crime category and the target crime category. We perform a element wise product between input weight vector μ and crime vector $\mathcal{CM}_{i,k}$ of region R_i in k-th time slot to generate a new vector which serve as the input to the recurrent framework. Similar operations are conducted between μ and $\mathcal{AM}_{i,k}$ for urban anomalies.

3.2 Hierarchical Recurrent Framework

We develop a hierarchical recurrent framework to encode the temporal dynamics of crime patterns and their inherent interrelations with urban anomalies. Recurrent Neural Network (RNN) models have been widely applied in time series analysis. There exist various RNN architectures with different recurrent units, such as RNN [23], Long Short Term Memory (LSTM) [43] and Gated recurrent units (GRU) [37]. GRU is similar to LSTM as both utilize gating information to prevent vanishing gradient problem in conventional RNN, but is computationally more efficient and effective on less training data [6]. Therefore, in this section, we introduce GRU as a concrete example of a recurrent unit for our recurrent framework. Our recurrent framework is flexible to employ other recurrent units (*e.g.*, LSTM). The effect of recurrent unit selection on model performance is explored in Section 4.

Our recurrent framework has a three-level GRU architecture. In particular, its first level *Crime-GRU* encoder, encodes the temporal dependencies of the time-ordered crime sequence $\mathcal{CM}_i$ of region R_i. In addition, the second level *Anomaly-GRU* encoder, models the time-ordered anomaly sequence $\mathcal{AM}_i$ of region R_i in a similar way.

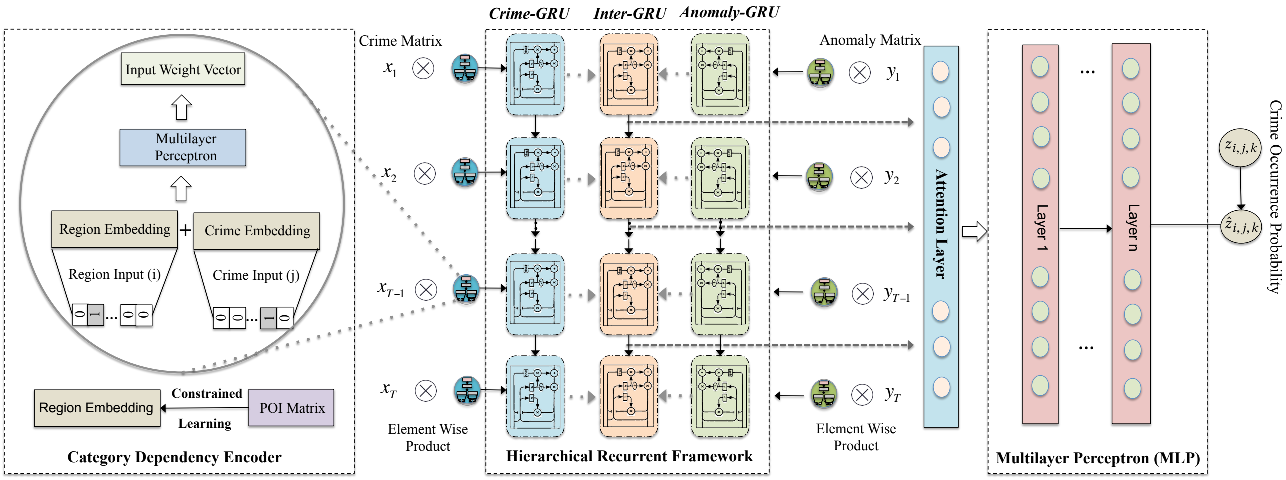


Figure 2: The DeepCrime Framework.

In the third level, we aim to employ another GRU encoder *Inter-GRU* to model the inherent dependencies between the occurrence of crimes and urban anomalies by concentrating their respective hidden state from each time slot.

In particular, GRU proposes to derive the vector representations of hidden states h_t for each time step t as follows:

$$\begin{aligned} x'_t &= x_t \odot ReLU(W_a[E_o; E_r] + b_a) \\ r_t &= \sigma(W_r h_{t-1} + V_r x'_t + b_i) \\ z_t &= \sigma(W_z h_{t-1} + V_z x'_t + b_o) \\ \widetilde{h}_t &= \phi(W_h h_{t-1} + V_h(x'_t \odot \widetilde{h}_{t-1}) + b_h) \\ h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \widetilde{h}_t \end{aligned} \tag{2}$$

where $W_* \in \mathbb{R}^{d_s \times d_s}$ represents the transformation matrix from the previous state h_{t-1} to GRU cell and $V_* \in \mathbb{R}^{d_x \times d_s}$ are the transformation matrices from input to GRU cell, where d_x and d_s denote the dimension of input vectors and hidden states, respectively. Furthermore, $b_* \in \mathbb{R}^{d_s}$ is defined as a vector of bias term. $\sigma(\cdot)$ and $\phi(\cdot)$ represent the sigmoid and tanh function, respectively. The $\odot$ operator denotes the element-wise product. In Eq. 2, r_t and z_t represent the update and reset gate, respectively. For simplicity, we denote Eq. 2 as $h_t = \text{GRU}(*, h_{t-1})$ in the following subsections. We formally define the corresponding hidden state in Crime-GRU, Anomaly-GRU and Inter-GRU, respectively as follows:

$$\begin{aligned} h_t &= \text{GRU}(CM_i, h_{t-1}) \\ g_t &= \text{GRU}(AM_i, g_{t-1}) \\ \lambda_t &= \text{GRU}(\Lambda, h_{t-1}) \end{aligned} \tag{3}$$

where h_t and g_t represent the hidden state corresponding to Crime-GRU and Anomaly-GRU encoder, respectively. In particular, we feed a concatenated vector $[h_t; g_t]$ as input into the Inter-GRU encoder to explore the dynamic interactions between the occurrences of crimes and urban anomalies. Λ denotes the sequence of concatenated vector across T time slots. Formally, $\Lambda = [[h_1; g_1], ..., [h_T; g_T]]$. λ_t is the hidden state of Inter-GRU encoder which captures the inherent dependencies between the crime and anomaly sequence.

3.3 Attention Mechanism

One limitation of the recurrent neural network based architectures lies in that they encode the input sequence to a fixed length internal representation, which results in worse performance for long input sequences [25]. To overcome this limitation, the attention mechanism was proposed to allow the proposed hierarchical recurrent framework to learn where to pay attention in the input sequence for each item in the output sequence [29]. Particularly, the attention mechanism aims to free the encoder-decoder architecture from the fixed-length internal representation by introducing a context vector to model the relevance. Formally, the attention mechanism can be represented as follows:

$$\begin{aligned} u_m &= tanh(W_v v_m + b_v) \\ \alpha_m &= \frac{exp(W_u u_m)}{\sum_{m'} exp(W_u u_{m'})} \\ \hat{v} &= \sum_{m=1}^{M} \alpha_m v_m \end{aligned} \tag{4}$$

where we use S to represent the attention dimension size. $W_v \in R^{S \times E}$ and $W_u \in R^{1 \times S}$ represent attention weight metrics. $b_m \in R^S$ is the attention bias. The number of input vectors is denoted by M. α_m indicates the learned importance weight which corresponds to projected vector v_m and $\hat{v}$ represents the new hidden representation which concatenates different hidden vectors. We further define W_m and W_u as two transmission matrices. For simplicity, we denote Eq. 4 as $\hat{v} = \text{Attention}(v_1, ..., v_m, ..., v_M)$ in the following subsections.

Our objective is to predict the crime occurrence in each region $R_i \in [R_1, ..., R_I]$ in the target time slot, based on the distribution patterns of past crimes and urban anomalies, *i.e.*, $[x_1,...,x_T]$ and $[y_1,...,y_T]$. However, our developed recurrent framework only encodes the input sequence from previous slots with a fixed length T using internal representations and the performance will drop when the sequence length is large. To address this drawback, we propose to employ an attention mechanism on our recurrent framework to capture the relevance of crime patterns learned from previous time slots in assisting the prediction of future crime occurrences.

In our attention mechanism, we estimate the importance of anomaly occurrence in past time slots by deriving a normalized importance weight via a softmax function. We define $\hat{\lambda}$ as the concatenated hidden state in our attention mechanism as: $\hat{\lambda} = \text{Attention}(\lambda_1, ..., \lambda_T)$.

3.4 Multilayer Perceptron (MLP)

Finally, the Multilayer Perceptron (MLP) component is utilized to derive the occurrence probability by capturing the non-linear dependencies between elements in hidden vectors. Formally, we represent MLP as follows:

$$
\begin{aligned}
L_1 &= \phi(W_1 \cdot \lambda_1 + b_1) \\
&\ldots\ldots \\
L_n &= \phi(W_n \cdot \lambda_n + b_n) \\
z_{i,j,k} &= \sigma(W' \cdot L_n + b') \qquad (5)
\end{aligned}
$$

where n represents the number of hidden layers in MLP (indexed by l). For the L_l layer, W_l and b_l represent the activation function (i.e., *ReLU* function) of MLP layers, weight matrix and bias vector, respectively. We further specify the activation function as *sigmod* (denoted as σ) to output the crime occurrence probability of category C_j at region R_i in k-th time slot, *i.e.*, $z_{i,j,k}$. In the experiments, we set the number of layers in MLP as 3.

3.5 Learning Process of DeepCrime

In this subsection, we describe the learning process of our DeepCrime framework as introduced in Section 3, our objective is to derive the value of $z_{i,j,k}$ which denotes: does crime with category C_j occur at region R_i in k-th time slot. A commonly used metric in the loss function of binary classification tasks is cross entropy [22]. Thus, we define our loss function which incorporates the constraint term in Eq. (1) as follows:

$$
\begin{aligned}
\mathscr{L} = &- \sum_{i,j,k} z_{i,j,k} \log \hat{z}_{i,j,k} + (1 - z_{i,j,k}) \log(1 - \hat{z}_{i,j,k}) \\
&+ \frac{1}{R} |E_R - PM \cdot W_{POI}| \qquad (6)
\end{aligned}
$$

where $\hat{z}_{i,j,k}$ denotes the estimated probability of j-th category crime occurrence in region R_i in k-th time slot. Here, S is the sampled crimes in the training process. The weights can be achieved by minimizing the loss function. In this work, we use Adaptive Moment Estimation (Adam) [17] to learn the parameters of DeepCrime.

4 EVALUATION

In this section, we perform experiments to evaluate the performance of *DeepCrime* on the real-world datasets collected from New York City (NYC). In particular, we aim to answer the following questions:

- **Q1**: How does our *DeepCrime* framework perform as compared to the state-of-the-art techniques in predicting crime occurences of different categories?
- **Q2**: Does *DeepCrime* consistently outperform other baselines in terms of prediction accuracy *w.r.t* different time windows with different training and testing time periods?

Table 1: Details of the datasets.

Data Source	**New York City Crime Reports**			
Time Span	From Jan, 2014 to Dec, 2014			
Category	Burglary		Robbery	
Number of Instances	16,720		16,557	
Category	Felony Assault		Grand Larceny	
Number of Instances	19,059		51,577	
Data Source	**Point-of-Interests (POI)**			
Category	#	Category	#	
Arts & Entertainment	720	Automotive & Vehicles	1505	
Business to Business	3717	Computers & Technology	637	
Education	1062	Food & Dining	3385	
Government & Community	3116	Health & Beauty	4336	
Home & Family	3616	Legal & Finance	1782	
Real Estate & Construction	4675	Shopping	1874	
Sports & Recreation	384	Others	1378	
Data Source	**311 Public-Service Complaints**			
Time Span	From Jan, 2014 to Dec, 2014			
Category	Noise		Blocked Driveway	
Number of Instances	151,174		92,335	
Category	Illegal Parking		Building Use	
Number of Instances	69,100		27,724	

- **Q3**: How is the performance of *DeepCrime* variants with different combinations of key components in the joint framework?
- **Q4**: How the different choices of model parameters (*e.g.*, embedding size and number of hidden layers) affect the performance of *DeepCrime*?
- **Q5**: How is the interpretation of our *DeepCrime* framework in capturing the unknown temporal relevance when predicting crimes.

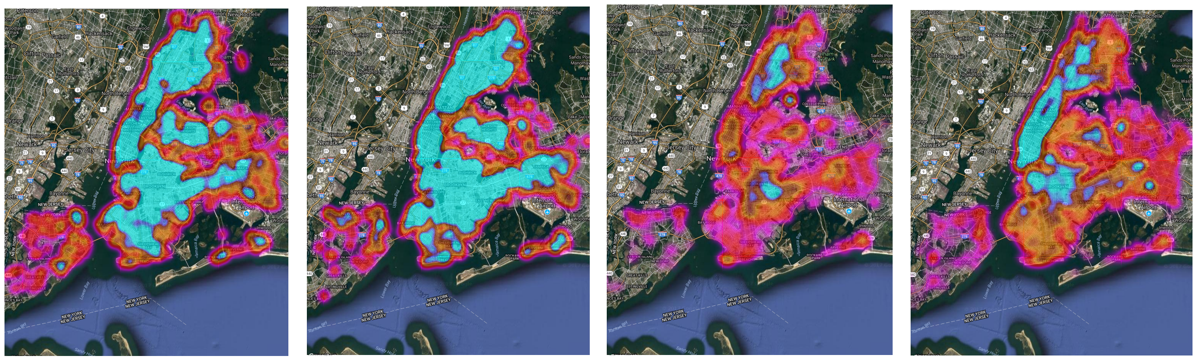
4.1 Experimental Setup

4.1.1 **Data.** We evaluated our framework with three datasets collected in New York City (NYC): (Detailed in Table 1).

1) **Crime Data**: We evaluated our *DeepCrime* framework on the real-world crime data collected from New York City (NYC) OpenData portal [1] from Jan 1, 2014 to Dec 31, 2014. Each crime record is in the format of (crime category, latitude, longitude, timestamp). In this work, we focus on the crime categories whose average frequency of occurrence is greater than 9 times for each region in a city per month, namely, *Robbery, Burglary, Felony Assault and Grand Larceny*.

Figure 3 shows the geographical distributions of different categories of crime occurrences in New York City (NYC) on August and December, respectively. From these visualization results, we can observe that (i) different geographical regions have different crime occurrence distributions given a specific crime category; (ii) crimes of different categories exhibit different occurrence patterns in the same region of a city; (iii) crimes from different time periods show different geographical distribution patterns. Inspired by the above observations, we propose to explicitly explore the inherent correlations between regions, crime categories and time slots with

[1] https://data.cityofnewyork.us/



(a) Burglary – Aug (b) Robbery – Aug (c) Felony – Dec (d) Grand – Dec

Figure 3: Geographical distribution of crime occurrences with different categories in New York City on August and December, respectively.

an attention-based hierarchical recurrent networks.

2) **Point of Interests (POIs)**: We collected 24,031 POIs of 14 categories (*e.g.*, Arts & Entertainment and Shopping, detailed in Table 1). Each POI is formatted as (venue name, category, address, latitude, longitude).

3) **311 Public Service Complaint Data**: These datasets are collected from 311 Service which documents urban complaint reports of different categories from citizens through a mobile app or phone calls. Each complaint record is in the format of (complaint category, latitude, longitude, timestamp). We selected 4 key complaint categories (*e.g.*, Noise, Blocked Driveway, Illegal Parking and Building Use) which are studied in [33].

We divided New York City into 77 disjointed geographical regions based on the information of political districts [2]. Each region is an area of the city as defined for police purposes. This method was also applied in geographical partition of the city in previous crime analysis work [26].

4.1.2 **Parameter Settings.** The hyper-parameters play important roles in *DeepCrime*, as they determine how the model will be trained. In our experiments, we vary each of the key parameters in *DeepCrime* and fix others to examine the parameter sensitivity of the proposed method. We implemented our framework based on TensorFlow and used Adam [17] as our optimizer to learn the model parameters. The hyperparameter settings are optimized with the grid search strategy. In our experiments, we set the batch size as 64, learning rate as 0.001 and the number of hidden layers in Multilayer Perceptron component as 3.

4.1.3 **Baselines.** We compare *DeepCrime* with the following four types of baselines: (i) variant of Recurrent Neural Network models for time series prediction. (*i.e.*, GRU); (ii) conventional time series forecasting methods (*i.e.*, ARIMA and SVR); (iii) both the conventional and neural feature-based supervised learning methods for classification (*i.e.*, LR, MLP and Wide&Deep); (iii) tenor factorization-based method for predictive analytics (*i.e.*, TriMine).

- **Support Vector Regression (SVR)** [3]: a non-parametric machine learning method for regression based on kernel functions.
- **Auto-Regressive Integrated Moving Average (ARIMA)** [16]: a well-known time series prediction model for understanding and predicting future values in a time series.
- **Logistic Regression (LR)** [13]: a statistical model which forecasts a region's crime occurrence based on temporal features (e.g., the day of a week and the month of a year) extracted from historical crime logs.
- **Multilayer Perceptron (MLP)** [7]: it incorporates temporal features from historical distributions of crimes into a deep neural network architecture, to model the non-linearities in crime data.
- **Tensor Decomposition (TriMine)** [24]: We apply this method to predict crime occurrences by extending the Matrix Factorization scheme to consider the temporal dimension of crime data. Specifically, we utilize a three-dimension tensor to represent the crime series of all regions in a city (1^{th} dimension–region, 2^{nd} dimension–crime category and 3^{rd} dimension–time).
- **Wide and Deep Learning (Wide&Deep)** [4]: a wide & deep learning framework to combine the strengths of wide linear models and deep neural networks for predictive analytics.
- **Gated Recurrent Unit (GRU)** [5]: a gating recurrent neural network model which has fewer parameters than LSTM by lacking an output gate to achieve computational efficiency.

In our experiments, all parameters are also learned using the Adam optimizer.

4.1.4 **Evaluation Protocols.** To validate the performance of all compared methods in predicting crime occurrences (posed as a classification problem) of each region in a city, we adopt two types of evaluation metrics: (i) we use *F1-score* (trade-off between precision and recall) to evaluate the accuracy of predicting a specific category of crime occurrence. (ii) We use *Marco-F1* and *Micro-F1* [12] to evaluate the prediction accuracy across different crime categories. These metrics have been widely used in the problems of multi-class classification to calculate the overall performance across different classes. In our work, we consider each crime category (*e.g.*, burglary and robbery) as an individual class. Note that a higher Micro-F1 and Macro-F1 score indicates better performance.

In our evaluation, we split the datasets chronologically into training (6.5 months), validation (0.5 month) and test (1 month) sets. The validation datasets are used to tune hyper-parameters and test datasets are used to evaluate the performance of all compared

[2] https://data.cityofnewyork.us/Public-Safety/Police-Precincts/78dh-3ptz/data

Table 2: Crime prediction results across different categories in terms of *Macro-F1* and *Micro-F1*.

Month	August		September		October		November		December	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1
SVR	0.6251	0.5202	0.6315	0.5244	0.6383	0.5380	0.6312	0.5400	0.6394	0.5457
ARIMA	0.6262	0.5377	0.6279	0.5345	0.6362	0.5514	0.6281	0.5478	0.6269	0.5451
LR	0.6341	0.5161	0.6348	0.5176	0.6378	0.5304	0.6260	0.5199	0.6307	0.5248
MLP	0.6432	0.5264	0.6492	0.5444	0.6482	0.5436	0.6389	0.5317	0.6407	0.5317
TriMine	0.6508	0.5326	0.6388	0.5141	0.6432	0.5258	0.6434	0.5538	0.6402	0.5335
Wide&Deep	0.6356	0.5209	0.6390	0.5251	0.6467	0.5419	0.6326	0.5366	0.6431	0.5464
GRU	0.6499	0.5836	0.6486	0.5803	0.6530	0.5879	0.6316	0.5659	0.6354	0.5720
DeepCrime	**0.6820**	**0.6200**	**0.6790**	**0.6227**	**0.6836**	**0.6233**	**0.6657**	**0.6009**	**0.6683**	**0.6110**

Table 3: Crime prediction results for individual category in terms of *F1-score*.

Category	Burglary					Robbery				
Algorithm	August	September	October	November	December	August	September	October	November	December
SVR	0.4661	0.4629	0.4921	0.4896	0.5241	0.4972	0.5094	0.5152	0.5201	0.5367
ARIMA	0.4767	0.4920	0.4961	0.4850	0.5234	0.5156	0.4967	0.5445	0.5333	0.5441
LR	0.4873	0.4941	0.4927	0.5032	0.5246	0.5165	0.4658	0.5066	0.5032	0.5246
MLP	0.4945	0.5106	0.5082	0.5087	0.5633	0.5514	0.5304	0.5586	0.5483	0.5537
TriMine	0.5081	0.4638	0.5110	0.5276	0.5306	0.5712	0.5096	0.5408	0.5161	0.5576
Wide&Deep	0.4642	0.5080	0.5236	0.4985	0.5482	0.5314	0.4878	0.5269	0.5325	0.5549
GRU	0.5394	0.5569	0.5633	0.5147	0.5378	0.5631	0.5446	0.5784	0.5491	0.5684
DeepCrime	**0.6173**	**0.6052**	**0.6051**	**0.5902**	**0.5912**	**0.6300**	**0.5848**	**0.6177**	**0.5993**	**0.6228**
Category	Felony Assault					Grand Larceny				
Algorithm	August	September	October	November	December	August	September	October	November	December
SVR	0.5891	0.6089	0.5750	0.5842	0.5893	0.8593	0.8581	0.8653	0.8426	0.8354
ARIMA	0.5967	0.5974	0.5746	0.5821	0.5627	0.8491	0.8542	0.8552	0.8366	0.8301
LR	0.5779	0.5992	0.5808	0.5704	0.5512	0.8713	0.8644	**0.8669**	0.8440	0.8405
MLP	0.5917	0.6272	0.5973	0.5961	0.5600	0.8728	0.8638	0.8650	0.8432	0.8423
TriMine	0.6282	0.6325	0.5891	0.6144	0.5913	0.8702	0.8636	0.8658	0.8442	0.8415
Wide&Deep	0.5865	0.6073	0.5964	0.5743	0.5675	0.8718	0.8633	0.8651	0.8430	0.8436
GRU	0.5992	0.6060	0.5909	0.5815	0.5561	0.8656	0.8529	0.8566	0.8356	0.8330
DeepCrime	**0.6459**	**0.6636**	**0.6336**	**0.6246**	**0.6120**	**0.8734**	**0.8645**	0.8664	**0.8443**	**0.8432**

algorithms. All experiments are conducted across 30 consecutive days in test time frames and the average performance is reported.

4.2 Performance Validation (Q1 and Q2)

To investigate the performance of all compared methods on different targeted time frames, we show the evaluation results from Aug 2014 to Dec 2014. We have the following key observations.

(1). Tables 2 and Table 3 list the evaluation results of all compared methods with respect to different training and test time windows. We observe that *DeepCrime* outperforms other baselines over different time frames (*i.e.*, from Aug to Dec). For example, *DeepCrime* achieves relatively 4.5% and 6.1% improvements over the best performed baseline (*i.e.*, GRU) in terms of Macro-F1 and Micro-F1 on October. In addition, although different time windows reflect a spectrum of temporal diversity which is maintained by month and season variation (*e.g.*, Aug–Summer, Sep, Oct–Autumn and Nov, Dec–Winter), our proposed method consistently achieves the best performance by capturing this temporal dynamic. Therefore, the evaluation results across different time frames demonstrate the effectiveness of *DeepCrime* in modeling time-evolving dependencies in crime sequences and reasonably interprets the importance of past crime occurrences in predicting future crimes.

(2). We perform experiments to evaluate *DeepCrime* in predicting individual crime categories as shown in Table 3. Overall, our proposed framework achieves the best performance across different crime categories in all cases. On average, *DeepCrime* achieves relatively 11.0%, 18.8% and 18.7% performance gains in terms of F1-score over GRU, Wide&Deep and TriMine, respectively (representing different types of baselines) when predicting Burglaries. In addition, obvious average improvements can also be obtained by *DeepCrime* in predicting Robberies, *e.g.*, 21.9%–LR. Another important observation is that the performance gain between *DeepCrime* and other baselines becomes larger as data becomes sparser (as shown in Table 1). This observation suggests that *DeepCrime* is capable of handing sparse crime data by exploring inherent region-category-time interactions and utilizing various ubiquitous data. In the occasional cases that *DeepCrime* misses the best performance in predicting Grand Larcenies, it still achieves competitive prediction results.

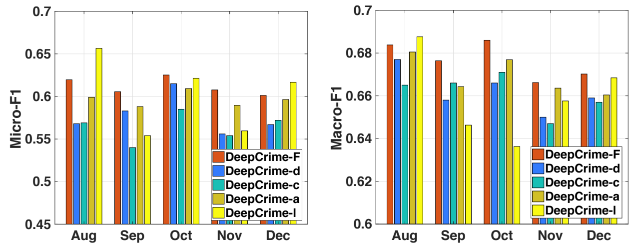
(3). We observe that *DeepCrime* shows improvement over all baselines. *First*, the large performance gap between *DeepCrime* and recurrent neural network-based scheme (*i.e.*, GRU) indicates the limitation of those approaches–only modeling the sequential pattern of the crimes and ignoring the relevant ubiquitous data and inherent region-category interactions. *Second*, the evaluation results shed light on the limitations of feature-based learning algorithms (*i.e.*, LR, MLP and Wide&Deep) which ignore the temporal dynamics embedded in the crime series data. *Third*, the significant performance improvement between *DeepCrime* and matrix factorization-based method (*i.e.*, TriMine) stem from explicitly modeling temporal dynamics of latent factors underlying crime occurrences with ubiquitous Data. *Fourth*, *DeepCrime* outperforms conventional time series forecasting methods (*i.e.*, ARIMA and SVR) due to their assumption of fixed temporal pattern.

4.3 Evaluations on Variants of *DeepCrime* (Q3)

In addition to comparing *DeepCrime* with state-of-the-art techniques, we also aim to get a better understanding of the proposed framework and evaluate the key components of *DeepCrime*. Namely, we aim to answer the following three questions: (1) Whether each key learning component plays a crucial role in the joint representation learning model *DeepCrime*? (2) Are the features extracted from other ubiquitous data (*e.g.*, POIs and urban anomalies) helpful for predicting crimes? (3) How does the selection of the recurrent unit affect the performance of *DeepCrime*? Hence, in our evaluation, we consider four variants of the proposed method corresponding to different analytical aspects mentioned above. We define the full version of *DeepCrime* introduced in Section 3 as *DeepCrime-F*.

- **Ubiquitous Data** *(DeepCrime-d)*: A simplified version of *DeepCrime* which models crime series data using a single level of gated recurrent units, *i.e.*, without incorporating the features extracted from other ubiquitous data into the solution.
- **Categorical Dependencies** *(DeepCrime-c)*: This method ignores the inherent dependencies between different crime categories, *i.e.*, using crime vectors from individual categories as the input to the proposed model.
- **Attention Mechanism** *(DeepCrime-a)*: Another simplified version of *DeepCrime* which only uses the hidden states generated by hierarchical gated recurrent units to predict the future crime occurrence, *i.e.*, without employing any attention mechanism.
- **Recurrent Unit Selection** *(DeepCrime-l)*: It utilizes the LSTM as the basic recurrent unit for encoding the crime and ubiquitous data with temporal dynamics in the proposed hierarchical architecture, *i.e.*, employing LSTM as the recurrent unit instead of GRU in DeepCrime for modeling the chronological sequence generated from both crime and anomaly data.

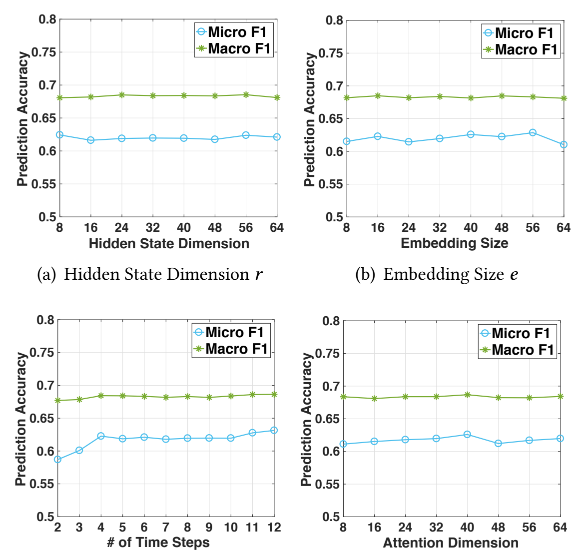
We report the evaluation results in Figure 4. From this figure, we can notice that the full version of our developed framework *DeepCrime-F* achieves the best performance in all cases. In particular, we summarize four key observations:



(a) Micro-F1 (b) Macro-F1

Figure 4: Evaluation on the Variants of DeepCrime in terms of *Micro-F1* and *Macro-F1*.

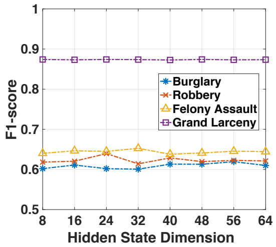
- (i) *DeepCrime-F* outperforms DeepCrime-d (without using the anomaly and POI data) in all cases, suggesting that ubiquitous data provides additional information in modeling the dynamic patterns of crime occurrences across time frames.
- (ii) Overall, *DeepCrime-F* outperforms the variant DeepCrime-c which does not consider the implicit dependencies across time slots between crimes with different categories. This observation justifies the effectiveness of *DeepCrime* in capturing the categorical interrelations between crime occurrences over time.
- (iii) Note that when the attention mechanism is applied to model the unknown temporal relevance between crime occurrences across time slots, the prediction performance is improved. Hence, we can see the efficacy of attention mechanism for helping *DeepCrime* make correct predictions.
- (iv) *DeepCrime-F* outperforms DeepCrime-l (LSTM unit) in most cases. We chose GRU in *DeepCrime* framework for improving computational efficiency and prediction accuracy.



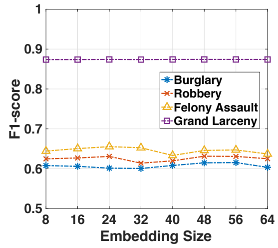
(a) Hidden State Dimension r (b) Embedding Size e

(c) # of Time Steps T (d) Attention Dimension Size S

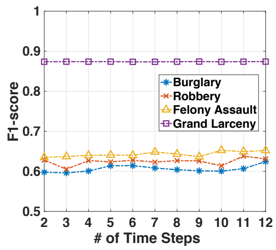
Figure 5: Parameter sensitivity study on the performance of *DeepCrime* in terms of *Micro-F1* and *Macro-F1* on August.



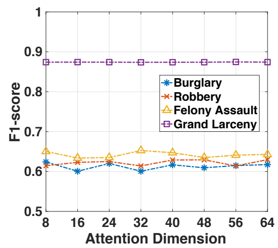
(a) Hidden State Dimension r



(b) Embedding Size e



(c) # of Time Steps T



(d) Attention Dimension Size S

Figure 6: Parameter sensitivity study on the performance of *DeepCrime* in terms of *F1-score* on August.

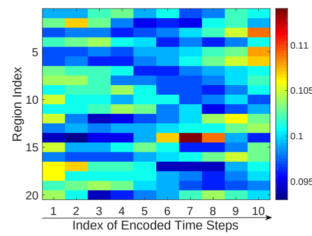
4.4 Parameter Sensitivity Studies (Q4)

DeepCrime involves several parameters (*i.e.*, embedding size of region, categorical crime representations, # of time steps and hidden state dimension in recurrent framework, dimension size in attention mechanism). To investigate the robustness of the *DeepCrime* framework, we examine how the different choices of parameters affect the performance of *DeepCrime* in predicting crimes. Except for the parameter being tested, we set other parameters at the default values.

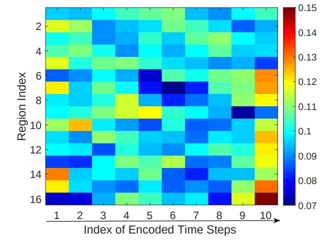
Figure 5 shows the evaluation results across crime categories (measured by Macro-F1 and Micro-F1) as a function of one selected parameter when fixing others. Overall, we observe that *DeepCrime* is not strictly sensitive to these parameters. We observe that the increase of prediction performance saturates as the length of input crime sequence reaches around 4. In addition, we set the embedding size as 32 in our experiments due to the balance between efficacy and computational cost, *i.e.*, the smaller the embedding size is, the more efficient the training process will be. Following the same experimental procedure, we further study the parameter sensitivity of *DeepCrime* in predicting individual categories as measured by F1-score and report the evaluation results in Figure 6. Similarly, we can observe that hyper-parameters have a relatively low impact on the performance of *DeepCrime*, which demonstrates the robustness of our proposed framework.

4.5 Case Study (Q5)

In addition to the above quantitative analysis, we further investigate our approach by performing a case study. We provide qualitative examples in Figure 7 for the better understanding of our attention model. In particular, Figure 7(a) and Figure 7(b) shows the attention weights of all true positive regions cases with respect to the Burglary prediction on Sep 09 (weekday) and the Robbery prediction on Nov 22 (weekend), respectively. From these figures, we can observe that the attention weights are time-evolving across the encoder time steps, which suggests that our *DeepCrime* is able to capture the dynamic patterns of crime distributions across time slots. Recall that attention weights represent the relevance weights of crime occurrences across previous time slots for making predictions.



(a) Attention weights of Burglary prediction



(b) Attention weights of Robbery prediction

Figure 7: Visualization of attention weights produced by *DeepCrime.*

5 RELATED WORK

Numerous novel urban sensing applications have been developed recently [14, 20, 21, 28, 32, 34, 35, 42]. For example, Lian *et al.* studied the problem of restaurant survival prediction by considering geographical information and user mobility [20]. Furthermore, Wang *et al.* proposed to spot and trace the latent trip purposes of taxi trajectories from a city [28]. However, the crime prediction problem in urban sensing remains a challenging problem to be solved. In this paper, we develop an end-to-end model to predict the future crime occurrence of each geographical region in a city.

There exist prior studies on crime rate inference and detecting crime hotspots [26, 36]. For example, Wang *et al.* aimed to infer crime rate in a city by utilizing Point-of-Interest information [26]. Yu *et al.* developed a boosting-based clustering algorithm to identify crime hotspots [36]. Our work is closely related to works that study the problem of crime prediction [8, 10, 11, 40] which can be categorized into two groups. (i) *statistical methods*: census statistical information was used to discuss crime events, such as demographic information [8] and symbolic racism [11]. (ii) *data mining techniques*: Gerber *et al.* using Twitter data to predict crimes in a city [10]. Zhao *et al.* addressed the crime prediction problem by considering spatial-temporal correlations between regions [40].

Most of the above studies forecast the crimes using statistical or conventional data mining approaches. However, those previous crime prediction techniques relied on a good amount of high quality static demographic data or ignored the dynamic temporal dependencies in the distributions of crime sequence. In contrast, this work develops a neural network-based crime prediction model which jointly models time-evolving dependencies in multi-dimensional crime data and incorporates both static and dynamic ubiquitous data (*i.e.*, POI and urban anomaly data) into our framework.

Our work is related to literature that focuses on modeling time-stamped data [15, 18, 30, 38, 39]. Recently, in light of the significant progress yielded by deep learning techniques on natural language processing and speech, many efforts have been made to apply recurrent neural networks (RNN) and its variants in modeling time series data [18, 23]. For example, Wu *et al.* predicted ratings of users for movies with an LSTM architecture by exploring users' historical behavioral trajectories [31]. Laptev *et al.* proposed a LSTM-based architecture for special event forecasting at Uber using heterogeneous time-series data [18]. Inspired by the above work, we have developed a new neural architecture to capture the time-varying patterns in crime sequences and implicit contextual signals embedded in relevant ubiquitous data.

6 CONCLUSION AND FUTURE WORK

Crime prediction is a challenging and important task, and interpreting the time-ordered sequential crime data is a hard and vital problem for predictive model in urban sensing. This paper explored the neural network architectures to explicitly model the evolving dependencies in time-ordered crime sequence and implicit multi-dimensional interactions between regions, categories and time slots. We evaluate our new framework on real-world datasets collected from NYC. The results showed that our approach achieves better performance when competing with baselines.

Notwithstanding the interesting problem and promising results, some directions exist for future work. *First*, our *DeepCrime* is a general framework to capture spatial-temporal-categorical dynamics, we would like to apply our method to a much broader set of urban data forecasting applications. *Second*, we would like to explore more data sources (*e.g.*, social media data) in addition to the ubiquitous data used in this work.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work was in part by the National Science Foundation (NSF) Grants IIS-1447795 and CNS-1629914, and National Natural Science Foundation of China (No. 61773324).

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*. Springer, 1–4.
[2] John Braithwaite. 1989. *Crime, shame and reintegration.* Cambridge University Press.
[3] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *TIST* 2, 3 (2011), 27.
[4] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, et al. 2016. Wide & deep learning for recommender systems. In *Recsys*. ACM, 7–10.
[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, et al. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).
[6] Rui Costa, Ioannis Alexandros Assael, Brendan Shillingford, Nando de Freitas, and TIm Vogels. 2017. Cortical microcircuits as gated-recurrent neural networks. In *NIPS*. 271–282.
[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Recsys*. ACM, 191–198.
[8] Coral Featherstone. 2013. Identifying vehicle descriptions in microblogging text with the aim of reducing or predicting crime. In *ICAST*. IEEE, 1–8.
[9] David Finkelhor. 2008. *Childhood victimization: Violence, crime, and abuse in the lives of young people.* oxford university Press.
[10] Matthew S Gerber. 2014. Predicting crime using Twitter and kernel density estimation. *Decision Support Systems* 61 (2014), 115–125.
[11] Eva GT Green, Christian Staerklé, and David O Sears. 2006. Symbolic racism and Whites' attitudes towards punitive and preventive crime policies. *Law and Human Behavior* 30, 4 (2006), 435–454.
[12] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. ACM, 855–864.
[13] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression.* Vol. 398. John Wiley & Sons.
[14] Chao Huang, Dong Wang, and Shenglong Zhu. 2017. Where are you from: home location profiling of crowd sensors from noisy and sparse crowdsourcing data. In *INFOCOM*. IEEE, 1–9.
[15] Chao Huang, Xian Wu, and Dong Wang. 2016. Crowdsourcing-based urban anomaly prediction system for smart cities. In *CIKM*. ACM, 1969–1972.
[16] Mehdi Khashei, Mehdi Bijari, and Gholam Ali Raissi Ardali. 2009. Improvement of auto-regressive integrated moving average models using fuzzy logic and artificial neural networks (ANNs). *Neurocomputing* 72, 4-6 (2009), 956–967.
[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[18] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. 2017. Time-series extreme event forecasting with neural networks at uber. In *ICML*.
[19] Jay Yoon Lee, U Kang, Danai Koutra, and Christos Faloutsos. 2013. Fast anomaly detection despite the duplicates. In *WWW*. ACM, 195–196.
[20] Jianxun Lian, Fuzheng Zhang, Xing Xie, and Guangzhong Sun. 2017. Restaurant Survival Analysis with Heterogeneous Information. In *WWW*. ACM, 993–1002.
[21] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. GeoMAN: multi-level attention networks for geo-sensory time series prediction. In *IJCAI*. 3428–3434.
[22] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. 2016. Large-Margin Softmax Loss for Convolutional Neural Networks.. In *ICML*. 507–516.
[23] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *KDD*. ACM, 1903–1911.
[24] Yasuko Matsubara, Yasushi Sakurai, Christos Faloutsos, Tomoharu Iwata, and Masatoshi Yoshikawa. 2012. Fast mining and forecasting of complex time-stamped events. In *KDD*. ACM, 271–279.
[25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*. 3104–3112.
[26] Hongjian Wang, Daniel Kifer, Corina Graif, and Zhenhui Li. 2016. Crime rate inference with big data. In *KDD*. ACM, 635–644.
[27] Hongjian Wang, Huaxiu Yao, Daniel Kifer, Corina Graif, and Zhenhui Li. 2017. Non-Stationary Model for Crime Rate Inference Using Modern Urban Data. In *TBD*. IEEE.
[28] Pengfei Wang, Yanjie Fu, Guannan Liu, Wenqing Hu, and Charu Aggarwal. 2017. Human Mobility Synchronization and Trip Purpose Detection with Mixture of Hawkes Processes. In *KDD*. ACM, 495–503.
[29] Xuejian Wang, Lantao Yu, Kan Ren, Guanyu Tao, Weinan Zhang, Yong Yu, and Jun Wang. 2017. Dynamic attention deep model for article recommendation by learning human editors' demonstration. In *KDD*. ACM, 2051–2059.
[30] Ami Wiesel, Ofir Bibi, and Amir Globerson. 2013. Time Varying Autoregressive Moving Average Models for Covariance Estimation. *TSP* 61, 11 (2013), 2791–2801.
[31] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *WSDM*. ACM, 495–503.
[32] Fei Wu, Pranay Anchuri, and Zhenhui Li. 2017. Structural Event Detection from Log Messages. In *KDD*. ACM, 1175–1184.
[33] Xian Wu, Yuxiao Dong, Chao Huang, Jian Xu, Dong Wang, and Nitesh V Chawla. 2017. UAPD: Predicting urban anomalies from spatial-temporal data. In *ECML/PKDD*. Springer, 622–638.
[34] Xian Wu, Yuxiao Dong, Baoxu Shi, Ananthram Swami, and Nitesh Chawla. 2018. Who will attend this event together? event attendance prediction via deep LSTM networks. In *SDM*. SIAM, 180–188.
[35] Huaxiu Yao, Fei Wu, Jintao Ke, and etc. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *AAAI*.
[36] Chung-Hsien Yu, Wei Ding, Ping Chen, and Melissa Morabito. 2014. Crime forecasting using spatio-temporal pattern with ensemble learning. In *PAKDD*. Springer, 174–185.
[37] Chuxu Zhang, Chao Huang, Lu Yu, Xiangliang Zhang, and Nitesh Chawla. 2018. Camel: content-aware and meta-path augmented metric learning for author identification. In *WWW*. ACM, 709–718.
[38] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction.. In *AAAI*. 1655–1661.
[39] Xuchao Zhang, Liang Zhao, Arnold P Boedihardjo, Chang-Tien Lu, and Naren Ramakrishnan. 2017. Spatiotemporal Event Forecasting from Incomplete Hyper-local Price Data. In *CIKM*. ACM, 507–516.
[40] Xiangyu Zhao and Jiliang Tang. 2017. Modeling Temporal-Spatial Correlations for Crime Prediction. In *CIKM*. ACM, 497–506.
[41] Xiangyu Zhao and Jiliang Tang. 2018. Crime in Urban Areas: A Data Mining Perspective. *SIGKDD Explorations* (2018).
[42] Guanjie Zheng, Susan L Brantley, Thomas Lauvaux, and Zhenhui Li. 2017. Contextual Spatial Outlier Detection with Metric Learning. In *KDD*. ACM, 2161–2170.
[43] Xiaodan Zhu, Parinaz Sobihani, and Hongyu Guo. 2015. Long short-term memory over recursive structures. In *ICML*. 1604–1612.