Seminar Report

on

THE TITLE OF SEMINAR

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Bachelor of Technology

IN

COMPUTER SCIENCE AND ENGINEERING

BY

Name of Student

University Register Number

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING KANNUR

Kannur District, Kerala State

November 2019

Institute Vision

"A globally renowned institution of excellence in engineering, education, research and consultancy."

Institute Mission

"To contribute to the society by providing quality education and training, leading to innovation, entrepreneurship and sustainable growth."

Department Vision

"To be a centre of excellence in the field of Computer Science & Engineering education and research, which extends its appreciated services to the industry and the society."

Department Mission

"To develop engineers with excellent analytic, design and implementation skills, who can expertise themselves as computer professionals, research engineers, entrepreneurs or as managers, while fulfilling their ethical and social responsibilities, in a globally competitive environment."

Department. of Computer Science and Engineering
GOVERNMENT COLLEGE OF ENGINEERING KANNUR
Kannur District, Kerala State

CERTIFICATE

Type date here

Certified that this is a bonafide record of the seminar paper presented by **Student name (university reg. no)** *with the title "*TITLE OF SEMINAR REPORT*" towards the partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Department of Computer Science and Engineering under A. P. J. Abdul Kalam Technological University during the year 2019-2020.*

Prof. Name of Guide	**Prof. Baburaj. K. V**	**Dr. Rafeeque. P. C**
Seminar Guide	Seminar Co-ordinator	HoD, Dept. of CSE

Acknowledgements

Type the acknowledgement here

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. Here, 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 is used. Furthermore, this framework is capable of automatically capturing the relevance of crime occurrences across different time periods. In particular, the 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 the framework over many competitive baselines across various settings.

Contents

List of Figures

List of Tables

Chapter 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. 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.

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:

1. **Temporal dynamics of crime pattern**

   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.

2. **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 crime categories over time.

3. **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.

4. **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, a neural network framework is proposed to predict the crime occurrences of different categories in each region of a city. First, a region-category interaction encoder is developed to handle the complex interactions between regions and categories of occurred crimes. Second, a hierarchical recurrent framework is proposed 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, the attention mechanism is used 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:

- A category dependency encoder is developed that jointly maps the region and crime into the same latent space with their time-evolving correlations preserved.

- A hierarchical recurrent framework is proposed 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.

- Extensive experiments are performed on real-world datasets collected from NYC. Evaluation results demonstrate that this framework significantly outperforms state-of-the-art baselines in terms of prediction accuracy across various settings.

Chapter 2

Literature Review

2.1 Urban sensing applications

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]. They explored various factors under the guidance of the following considerations: (H1) the geographical placement of the store play an important role in the store's operation; (H2) people's offline mobility patterns to the store as well as its nearby places influence the business; (H3) user's rating scores (e.g., on Yelp) are explicit evaluations of the store from the customers' point of view; (H4) besides well-formatted rating scores, review words contain more rich information which a simple numeric score does not cover.

Wang et al. proposed to spot and trace the latent trip purposes of taxi trajectories from a city [28]. They identified a very important property of human mobility, which is, human mobility synchronization. In other words, if two regions share similar spatial configurations and urban functions in a particular time period, the two regions are likely to have similar patterns of arrival events.

Wu et al. proposed an end-to-end Deep Event Attendance Prediction (DEAP) framework—a three-level hierarchical LSTM architecture—to explicitly model users' multi-dimensional and evolving preferences [34]. DEAP explores the rich contextual information of events to address the aforementioned event cold-start challenge. In specific, its first level transforms events' contextual information into latent embedding vectors in a non-linear way. In the second level, they aimed to encode the evolving exclusive preferences of users by considering their attendance behavior across different groups. In DEAP's third level, it encodes users' sequential preferences to capture the time-evolving attendance patterns and interacts with the generated embedding vectors from the first two levels. With the three-level LSTM architecture, the generated semantic embedding vectors encode multi-dimensional preferences (i.e., sequential, contextual, and exclusive preferences). Finally, the embeddings are fed into a Multilayer Perceptron (MLP) for predicting the event attendance of each user.

However, the crime prediction problem in urban sensing remains a challenging problem to be solved. Here, we develop an end-to-end model to predict the future crime occurrence of each geographical region in a city.

2.2 Crime rate inference and detecting crime hotspots

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]. POI data provides venue information such as GPS coordinates, category, popularity, and reviews. These POIs mostly belong to categories such as food, shop, transit, education, etc. Using such categorical information of POIs are useful to profile neighborhood functions. Such neighborhood functions could further help predict crime rate. Their experiments showed that incorporating POI features significantly improve the crime rate inference. Adding POI features in addition to demographics features reduced the relative error by at least 5% in their experiments. This demonstrated that POI data provides additional information about the communities that is not covered by the demographics

Yu et al. developed a boosting-based clustering algorithm to identify crime hotspots [36]. The main idea of this approach is to iteratively pick a set of local patterns which give the least classification error at each boosting round. Each set of local patterns is referred as an ensemble spatio-temporal pattern and is assigned a score. At the end of boosting, a global pattern is constructed from these ensemble patterns. This global pattern is capable of predicting crime by scaling the total score of an input, a collection of crime indicators, evaluated on each crafted ensemble patterns

2.3 The problem of crime prediction

This project is closely related to works that study the problem of crime prediction [8, 10, 11, 40] which can be categorized into two groups.

1. **Statistical methods:**

   Census statistical information was used to discuss crime events, such as demographic information [8] and symbolic racism [11].

2. **Data mining techniques:**

   Gerber et al. used Twitter data to predict crimes in a city [10]. They pursued three objectives: (1) quantify the crime prediction gains achieved by adding Twitter-derived information to a standard crime prediction approach based on kernel density estimation (KDE), (2) identify existing text processing tools and associated parameterizations that can be employed effectively in the analysis of tweets for the purpose of crime prediction, and (3) identify performance bottlenecks that most affect the Twitter-based crime prediction approach. Their results indicated progress toward each objective. They have achieved crime prediction performance gains across 19 of the 25 different crime types in their study using a novel application of statistical language processing and spatial modeling.
   Zhao et al. addressed the crime prediction problem by considering spatial-temporal correlations between regions [40]. They exploited temporal-spatial correlations for crime prediction with urban data. In essence, their aim was to investigate the following two challenging questions: (i) what temporal-spatial patterns can be observed about crimes with urban data; and (ii) how to model these patterns mathematically for crime prediction. For temporal-spatial patterns, they focused their investigation on (a) intraregion temporal correlation and (b) inter-region spatial correlation. Intra-region temporal correlation tells how crime evolves over time for a region in a city; while inter-region spatial correlation suggests the geographical influence among regions in the city. They proposed a novel framework TCP, which captures temporal-spatial correlations for crime prediction.

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 the framework.

This work is related to literature that focuses on modeling timestamped data [15, 18, 30, 38, 39]. Recently, in light of the significant progress yielded by deep learning techniques on natural language Session 9C: Neural Prediction CIKM'18, October 22-26, 2018, Torino, Italy 1431 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]. They proposed Recurrent Recommender Networks (RRN) that are able to predict future behavioral trajectories. This was achieved by endowing both users and movies with a LSTM autoregressive model that captures dynamics, in addition to a more traditional low-rank factorization. On multiple real-world datasets, their model offers excellent prediction accuracy and is very compact.

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, a new neural architecture was developed to capture the time-varying patterns in crime sequences and implicit contextual signals embedded in relevant ubiquitous data.

Chapter 3

Methodology

In this section, the details of the framework are presented. It consists of three major modules: *Region-Category Interaction Encoder*, *Hierarchical Recurrent Framework* and *Attention Mechanism.* These three modules will be explained in detail in the following subsections. The model architecture is presented in Figure 1, where the output of one layer serves as the input to the next one.

3.1 Category Dependency Encoder

To consider the geographical context of regions, the POI information is incorporated into the process of generating the region embedding vector. Formally, the constraint term is defined 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, the input weight vector μ is defined 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. An element wise product between input weight vector μ and crime vector $CM_{i,k}$ of region R_i in k-th time slot is performed to generate a new vector which serve as the input to the recurrent framework. Similar operations are conducted between μ and $AM_{i,k}$ for urban anomalies.

3.2 Hierarchical Recurrent Framework

A hierarchical recurrent framework is developed 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

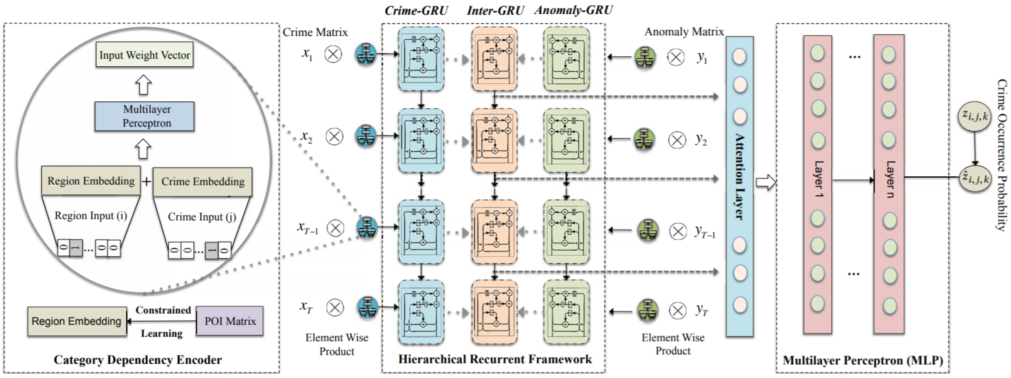


Figure 1: The model architecture of the framework

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, GRU is introduced as a concrete example of a recurrent unit for the recurrent framework. The 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

The 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 CM_i of region R_i. In addition, the second level *Anomaly-GRU* encoder, models the time-ordered anomaly sequence of region R_i in a similar way.

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

The corresponding hidden state in Crime-GRU, Anomaly-GRU and Inter-GRU, respectively are formally defined as follows:

$$\begin{aligned} h_t &= \mathrm{GRU}(CM_i, h_{t-1}) \\ g_t &= \mathrm{GRU}(AM_i, g_{t-1}) \\ \lambda_t &= \mathrm{GRU}(\Lambda, h_{t-1}) \end{aligned} \tag{2}$$

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:

$$\begin{aligned} u_m &= tanh(W_\nu \nu_m + b_\nu) \\ \alpha_m &= \frac{exp(W_u u_m)}{\sum_{m'} exp(W_u u_{m'})} \\ \hat{\nu} &= \sum_{m=1}^{M} \alpha_m \nu_m \end{aligned} \tag{3}$$

where we use S to represent the attention dimension size. $W_\nu \in R^{S\times R}$ 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 m indicates the learned importance weight which corresponds to projected vector ν_m and $\hat{\nu}$ represents the new hidden representation which concatenates different hidden vectors. W_u and W_m are defined as two transmission matrices. For simplicity, Eq. 3 is denoted as $\hat{\nu} = \text{Attention}(\nu_1, ..., \nu_m, ..., \nu_M)$ 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,the 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, an attention mechanism is employed on the recurrent framework to capture the relevance of crime patterns learned from previous time slots in assisting the prediction of future crime occurrences.

In this attention mechanism, the importance of anomaly occurrence in past time slots is estimated by deriving a normalized importance weight via a softmax function. $\hat{\lambda}$ is defined as the concatenated hidden state in the 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, the MLP is represented as follows:

$$\begin{aligned}
L_1 &= \phi(W_1 \cdot \lambda_1 + b_1) \\
&\ldots\ldots \\
L_n &= \phi(W_n \cdot \lambda_n + b_n) \\
z_{i,j,k} &= \sigma(W^{'} \cdot L_n + b^{'})
\end{aligned} \tag{4}$$

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. The activation function is specified 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, the number of layers in MLP is set as 3.

3.5 Learning Process

In this subsection, learning process of the framework as introduced in Section 3, the 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, the loss function which incorporates the constraint term in Eq. 1 is defined as follows:

$$\begin{aligned}
L = &-\sum_{i,j,k} z_{i,j,k} \log \hat{z}_{i,j,k} + (1 - z_{i,j,k}) \log(1 - \hat{z}_{i,j,k}) \\
&+ \frac{1}{R} |E_R - P_M \cdot W_{POI}|
\end{aligned} \tag{5}$$

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, Adaptive Moment Estimation (Adam) [17] is used to learn the parameters of the framework.

Chapter 4

Applications

Here, you may give data and information regarding the experimental setup and the simulation results. All the chapter titles should match your paper.

Chapter 5

Experimental Results

In this section, experiments are performed to evaluate the performance of the framework on the real-world datasets collected from New York City (NYC). In particular, the aim is to answer the following questions:

- **Q1**: How does the framework perform as compared to the state-of-the-art techniques in predicting crime occurences of different categories?
- **Q2:** Does the framework consistently outperform other baselines in terms of prediction accuracy *w.r.t* different time windows with different training and testing time periods?
- **Q3:** How is the performance of the framework 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 the framework?
- **Q5:** How is the interpretation of the framework in capturing the unknown temporal relevance when predicting crimes?

5.1 Experimental Setup

5.1.1 Data

The framework was evaluated with three datasets collected in New York City (NYC): (Detailed in Table 5.1).

1) **Crime Data**: The framework was valuated on the real-world crime data collected from New York City (NYC) Open-Data portal from Jan 1, 2014 to Dec 31, 2014. Each crime record is in the format of (crime category, latitude, longitude, timestamp). In this work, the focus is on the crime categories whose

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

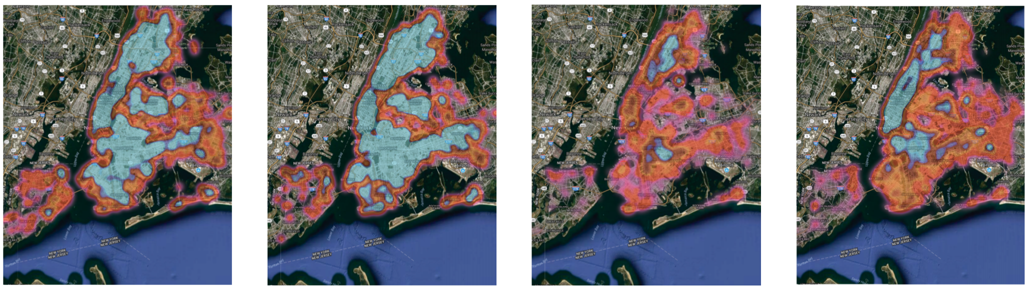
Data Source	**311 Public-Service Complaints**	
Time Span	From Jan, 2014 to Dec, 2014	
Category	Noise	Blocked Driveway
Number of Instances	151,174	92,335
Category	Illegal Parking	Building Use
Number of Instances	69,100	27,724

Table 5.1: Detail of the datasets

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 2 shows the geographical distributions of different categories of crime occurrences in New York City (NYC) on August and December, respectively. From these visualization results, it can be observed 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, the inherent correlations between regions, crime categories and time slots with an attention-based hierarchical recurrent networks is explicitly explored.

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



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

Figure 2: Geographical distribution of crime occurrences with different categories in New York City on August and December, respectively.

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).4 key complaint categories (*e.g.,* Noise, Blocked Driveway, Illegal Parking and Building Use) were selected which are studied in [33]. New York City was divided 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.

5.1.2 Parameter Settings

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

5.1.3 Baselines

The framework is compared 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]: this method is applied to predict crime occurrences by extending the Matrix Factorization scheme to consider the temporal dimension of crime data. Specifically, a three-dimension tensor is utilized to represent the crime series of all regions in a city (1^{st} 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 the experiments, all parameters are also learned using the Adam optimizer.

5.1.4 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, two types of evaluation metrics are adopted: (i) *F1-score* (trade-off between precision and recall) is used to evaluate the accuracy of predicting a specific category of crime occurrence. (ii) *Marco-F1* and *Micro-F1* [12] is used 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 this work, each crime category (*e.g.,* burglary and robbery) is considered as an individual class. Note that a higher Micro-F1 and Macro-F1 score indicates better performance.

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

5.2 Performance Validation (Q1 and Q2)

To investigate the performance of all compared methods on different targeted time frames, evaluation results from Nov 2014 and Dec 2014 are shown. The following key observations can be made.
(1) Table 5.2 and Table 5.3 list the evaluation results of all compared methods with respect to different training and test time windows. It can be observed that the proposed framework outperforms other baselines over different time frames (i.e., from Nov and Dec).In addition, although different time windows reflect a spectrum of temporal diversity which is maintained by month and season variation, the proposed method consistently achieves the best performance by capturing this temporal dynamic. Therefore, the evaluation results across different time frames demonstrate the effectiveness of the framework in modeling time-evolving dependencies in crime sequences and reasonably interprets the importance of past crime occurrences in predicting future crimes

Month	August		September	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1
SVR	0.6312	0.5400	0.6394	0.5457
ARIMA	0.6281	0.5478	0.6269	0.5451
LR	0.6260	0.5199	0.6307	0.5248
MLP	0.6389	0.5317	0.6407	0.5317
TriMine	0.6434	0.5538	0.6402	0.5335
Wide&Deep	0.6326	0.5366	0.6431	0.5464
GRU	0.6316	0.5659	0.6354	0.5720
Proposed framework	**0.6657**	**0.6009**	**0.6683**	**0.6110**

Month	October		November	
Algorithm	Macro-F1	Micro-F1	Macro-F1	Micro-F1
SVR	0.6312	0.5400	0.6394	0.5457
ARIMA	0.6281	0.5478	0.6269	0.5451
LR	0.6260	0.5199	0.6307	0.5248
MLP	0.6389	0.5317	0.6407	0.5317
TriMine	0.6434	0.5538	0.6402	0.5335
Wide&Deep	0.6326	0.5366	0.6431	0.5464
GRU	0.6316	0.5659	0.6354	0.5720
Proposed framework	**0.6657**	**0.6009**	**0.6683**	**0.6110**

Month	December	
Algorithm	Macro-F1	Micro-F1
SVR	0.6394	0.5457
ARIMA	0.6269	0.5451
LR	0.6307	0.5248
MLP	0.6407	0.5317
TriMine	0.6402	0.5335
Wide&Deep	0.6431	0.5464
GRU	0.6354	0.5720
Proposed framework	**0.6683**	**0.6110**

Table 5.2: Crime prediction results across different categories in terms of Macro-F1 and Micro-F1

Category	Burglary		Robbery	
Algorithm	November	December	November	December
SVR	0.4896	0.5241	0.5201	0.5367
ARIMA	0.4850	0.5234	0.5333	0.5441
LR	0.5032	0.5246	0.5032	0.5246
MLP	0.5087	0.5633	0.5483	0.5537
TriMine	0.5276	0.5306	0.5161	0.5576
Wide&Mine	0.4985	0.5482	0.5325	0.5549
GRU	0.5147	0.5378	0.5491	0.5684
Proposed framework	**0.5902**	**0.5912**	**0.5993**	**0.6228**
Category	Felony Assault		Grand Larceny	
Algorithm	November	December	November	December
SVR	0.5842	0.5893	0.8426	0.8354
ARIMA	0.5821	0.5627	0.8366	0.8301
LR	0.5704	0.5512	0.8440	0.8405
MLP	0.5961	0.5600	0.8432	0.8423
TriMine	0.6144	0.5913	0.8442	0.8415
Wide&Mine	0.5743	0.5675	0.8430	0.8436
GRU	0.6316	0.5659	0.5815	0.5561
Proposed framework	**0.6246**	**0.6120**	**0.8443**	**0.8432**

Table 5.3: Crime prediction results for individual category in terms of F1-score

Chapter 6

Conclusions and Future Scope

The main idea of this chapter is to give a brief conclusion of the work done or presented and aim more on the plan of future extensions of the work.

Bibliography