Traffic Anomaly Prediction Based on Joint Static-Dynamic Spatio-Temporal Evolutionary Learning

Xiaoming Liu, *Member, IEEE,* Zhanwei Zhang, Lingjuan Lyu, *Member, IEEE,* Zhaohan Zhang, Shuai Xiao, Chao Shen, *Member, IEEE*, Philip S. Yu, *Fellow, IEEE*

Abstract—Accurate traffic anomaly prediction offers an opportunity to save the wounded at the right location in time. However, the complex process of traffic anomaly is affected by both various static factors and dynamic interactions. The recent evolving representation learning provides a new possibility to understand this complicated process, but with challenges of imbalanced data distribution and heterogeneity of features. To tackle these problems, this paper proposes a spatio-temporal evolution model named SNIPER for learning intricate feature interactions to predict traffic anomalies. Specifically, we design spatio-temporal encoders to transform spatio-temporal information into vector space indicating their natural relationship. Then, we propose a temporally dynamical evolving embedding method to pay more attention to rare traffic anomalies and develop an effective attention-based multiple graph convolutional network to formulate the spatially mutual influence from three different perspectives. The FC-LSTM is adopted to aggregate the heterogeneous features considering the spatio-temporal influences. Finally, a loss function is designed to overcome the 'over-smoothing' and solve the imbalanced data problem. Extensive experiments show that SNIPER averagely outperforms state-of-the-arts by 3.9%, 0.9%, 1.9% and 1.6% on Chicago datasets, and 2.4%, 0.6%, 2.6% and 1.3% on New York City datasets in metrics of AUC-PR, AUC-ROC, F1 score, and accuracy, respectively.

Index Terms—anomaly prediction, spatio-temporal data, static-dynamic embedding, imbalanced data distribution, multiple graph convolutional network

1 Introduction

Motor vehicle collision is the first leading cause of death among people aged 15-29 years, and more than 50 million people suffer non-fatal injuries with many incurring a disability as a result of their injuries [1]. Thus, the demand for ensuring road safety becomes more urgent. Works on accurate prediction of traffic anomaly [2] can be of great help for decreasing the risk of traffic accidents by warning people involved in transportation systems the high risk of accident occurrence in particular locations ahead of time, in which discovering the dynamic patterns of the anomaly is essential. Recently, the various kinds of publicly available spatio-temporal datasets provide a new possibility for researchers to explore the complex dynamic process of traffic collisions, and find the intricate patterns to indicate the potential traffic risk. For example, traffic flow data estimated from public traffic usage reflects the vehicle density on the road (crowed traffic may increase risk of traffic accident); the weather data crawled from the Internet describes road conditions (snow would make the road frozen, which leads to more collisions); and traffic-related social media reports many traffic anomalies (an accident on the road will disturb traffic order and then increase the risk of collisions), etc. Now, an interesting problem arises, *i.e.*, could we predict the possible traffic anomalies based on these publicly available spatio-temporal datasets?

Imbalanced data distribution is one of the biggest obstacles for making accurate prediction [1], *i.e.*, the number of normal traffic events is far more than that of traffic anomalies. What's more, the complicated dynamic process of traffic anomalies is affected by various spatio-temporal factors, which makes traffic accidents hard to predict. Thus, although the traffic safety problem has attracted significant attention from the community, few focus on the anomaly prediction problem in traffic by using ample publicly available spatio-temporal datasets. Recently, Zhang *et al.* [2] propose a multitask deep-learning framework to predict the traffic flow, in which they employed convolutional neural networks (CNN) to effectively extract the grid-based region features. To model both spatial and temporal relations, Yao *et al.* [3] come up with a deep multi-view spatio-temporal network, which takes advantage of recurrent neural networks (RNN) to extract the temporal features by considering the temporal periodicity. Moreover, Wang *et al.* [4] propose the GSNet to learn the spatial-temporal correlations from the aspects of geographical and semantic for traffic accident risk prediction.

However, most existing methods fail to consider the

- *Xiaoming Liu and Chao Shen are the corresponding authors.*
- *X. Liu, Z. Zhang, Z. Zhang, and C. Shen are with School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China.*
  L. Lyu is with Sony AI, 1-7-1 Konan Minato-ku, Tokyo, 108-0075 Japan
  S. Xiao is a research scientist with Alibaba Group.
  P. S. Yu is with University of Illinois at Chicago, IL, USA.
- *E-mail: xm.liu@xjtu.edu.cn, {zwzhang, zzh1103}@stu.xjtu.edu.cn, lingjuanlvsmile@gmail.com, chaoshen@xjtu.edu.cn, shuai.xsh@alibaba-inc.com, psyu@uic.edu.*

1. http://www.who.int/mediacentre/factsheets/fs358/en/
2. In this paper, the traffic anomaly means the traffic collision event.

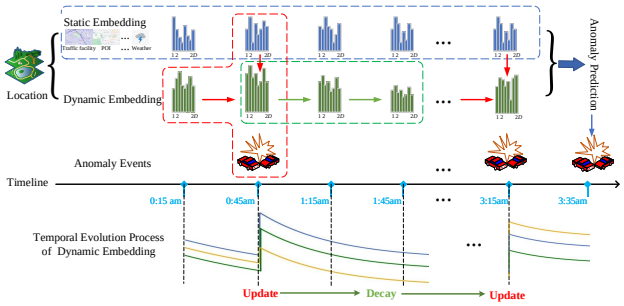


Fig. 1. Illustration of traffic anomaly event prediction based on joint static-dynamic spatio-temporal evolutionary learning. Traffic anomalies are predicted based on static embedding and dynamic embedding, which are obtained by a joint spatio-temporal evolutionary representation learning model. Red arrows denote **Update** operations on dynamic embedding in evolution progress, while green arrows denote **Decay** operations. The static embedding is the direct encoding of current traffic features which describe the stationary properties of current traffic events and locations, and are acquired directly by sensors or other installations, such as POIs, weather conditions and statistical traffic flow, etc.; while the dynamic embedding is evolving from the beginning time to the current time to encode the time-varying properties, which could capture mutually recursive correlation and dependency between the anomaly traffic events in timelines. The evolving process of dynamic embedding consists of the conditional update based on the static embedding and the continuous decay with time flowing. As shown in the red frame, the dynamic embedding is updated by aggregating the current static embedding of the location where traffic anomaly occurs at 0:45 am. And then the dynamic embedding decays when there exists no traffic anomaly, which is illustrated in the green frame. The dynamically evolving process is also indicated by the three lines under the timeline, which include the sudden increase and continuous decay. In this way, the dynamic embedding leverages the current static embedding and the historic dynamic information to depict the temporal dependencies among anomaly events.

following factors for predicting traffic anomalies, which are illustrated in Fig. 1. **First, the combination of static features and dynamic features.** As shown in Fig. 1, traffic anomalies are affected by static feature embeddings, while mutual influences among the events are changing dynamically with the time flowing. How to jointly learn the representation of static features and dynamic features is important to the prediction of traffic anomalies. However, many works [5], [6], [7], [8] only exploit the spatio-temporal data, such as Point Of Interests (POI), weather condition, and traffic flow etc., as static features for machine learning method to predict the possible anomalies. They ignore the dynamically interactive process among the locations and events, whose embedding features are temporally evolving and spatially mutual interacting. **Second, imbalanced data distribution.** As clarified earlier, compared with the normal traffic-related data, the traffic anomaly data is very sparse in both spatial and temporal spaces. It should be noted that though the traffic anomalies are sparse, there are always connections or traceable patterns among anomalies. Most existing methods [9], [10], [11], [12] ignore the phenomena and simply train their models on the imbalanced data, which hardly capture the inner pattern in the traffic anomalies. **Third, evolution strategy in the dynamical computation process.** Some latest works [13], [14], [15], [16] are proposed to depict the dynamic process of events by illustrating the event occurring rate in the space-time dimensions. But the imbalanced data distribution, especially the ones with few anomalies, makes it difficult to learn the representation of traffic anomalies for recent dynamical models. In other words, those algorithms lack the ability to apply the evolution strategy on updating dynamic features for traffic anomalies to capture the difference between the normal and anomaly.

To address this research gap, we are motivated to propose a joint **s**tatic-dy**n**amic spat**i**o-tem**p**oral **e**volutiona**r**y learning model (named SNIPER) to predict the traffic anomalies. The research in this paper takes a step towards collision prediction from the perspective of multiple publicly-available spatio-temporal data mining, which could enhance traffic safety and protect the urban from collision fatalities. The main contributions of this paper are summarized as follows:

- **Representation learning algorithm:** we propose a joint static-dynamic spatio-temporal feature representation learning algorithm to predict the traffic anomalies. In detail, we develop a temporally dynamic evolving feature embedding model and design a spatially mutual influence representation learning network. The two kinds of features are fused by one FC-LSTM model [17] with a fully connected layer by considering their spatio-temporal dependency.
- **Evolution strategy:** we design a conditional evolution strategy with computations of **Update** and **Decay** based on a dual spatio-temporal information encoder to distinguish anomalous and normal events and improve the training efficiency.
- **Loss function:** we design a dynamic loss function and combine it with an improved focal loss to address the issue of imbalanced data distribution by re-scaling the dynamical feature evolutionary process based on the historical collision information.
- **Outstanding performance:** the proposed algorithm SNIPER outperforms 5 baseline algorithms and 5 state-of-the-art algorithms in predicting the traffic anomalies based on two real-world large datasets. In detail, it averagely surpasses the state-of-the-art method with the best performance by 3.9%, 0.9%, 1.9% and 1.6% on Chicago datasets, and by 2.4%, 0.6%, 2.6% and 1.3% on New York City datasets in terms of AUC-PR, AUC-ROC, F1 score, and accuracy, respectively.

The codes and datasets are available on the website: *https://github.com/zwzhangzzz/SNIPER*.

2 Related Work

This section reviews the related works that are relevant to our work.

Anomaly Prediction. Anomaly could be defined as an observation that deviates considerably from some concept of normality [18], [19]. Recently, deep learning methods have been frequently used in the area of anomaly prediction, including autoencoder-based approaches [20], [21], [22] and GANs-based approaches [23], [24], [25]. However, these approaches concentrate much on reconstructing data instances, neglecting the nature of anomaly data. Anomaly prediction intends to forecast the rare object or unexpected

events in the future, which means positive samples take a small portion of the dataset. In this case, a model can hardly learn the anomalous pattern and easily overfit the training set. Zhang *et al.* [26] are inspired by Extreme Value Theory and propose a new form of loss called Extreme Value Loss (EVL) to handle extreme events in a fine-grained way. Ren *et al.* [27] borrow the idea of Spectral Residual (SR) [28] which is an efficient unsupervised algorithm first proposed to tackle visual saliency tasks. SR provides a solution on amplifying the significance of anomaly points. They apply CNN on the output of SR model and get satisfying results on the anomaly prediction problem in an unsupervised learning manner. Performing anomaly prediction on limited and imbalanced data is becoming a focused area of research. Pang *et al.* [29] apply a prior probability on deviation learning to enlarge statistically deviation between limited anomalies and normal data objects. However, few studies have been conducted for traffic anomaly prediction from a joint static-dynamic perspective.

Representation Learning. With the rapid development of Internet technology, more publicly available heterogeneous data [30] allows us to uncover the dynamic process of events, which can reveal the patterns and mutual influence among complicated events. Although there are some outstanding static graph embedding methods [31], [32], [33] that represent each node with a single vector, changes occurring on node embedding over a time series are neglected. To address the dynamic pattern, it is of great importance to learn the dynamic evolution representation of objects of interest. Trivedi *et al.* [34] come up with an inductive deep representation learning framework that could represent evolving information over dynamic graphs in the form of low dimensional node embedding. The learned embedding promotes communication and association processes between nodes over dynamic graphs. Goyal *et al.* [35] evaluate how graph dynamics influence the prediction performance and propose a model which consists of dense and recurrent layers to learn the temporal transitions in the dynamic graph. Kumar *et al.* [36] propose a coupled recurrent model to update the embedding at each interaction and future embedding trajectory of a user and an item in recommendation system. Zheng *et al.* [37] adapt an encoder-decoder architecture with multiple attention blocks to model the relation between historical and future time steps, which helps to alleviate error propagation and learn a precise and meaningful representation of graph nodes.

Graph Convolutional Network. To address pattern extraction problems on the non-Euclidean data structure which traditional convolution method cannot solve, Graph Convolutional Network (GCN) has been proposed and achieves astounding performance on different tasks based on graph structure. Yu *et al.* [38] propose STGCN which is able to capture spatial and temporal dependencies in mid-and-long term traffic forecasting problems. However, it lacks of the capability to model spatio-temporal correlations dynamically. Pan *et al.* [12] employ a sequence-to-sequence architecture, which consists of an encoder to encode the historical information and a decoder to make predictions in chronological order. Wu *et al.* [39] propose a novel spatio-temporal graph convolutional mechanism named Graph WaveNet. They capture the heterogeneous feature by a novel dependency matrix during different time periods. Geng *et al.* [40] encode pair-wise correlations among regions in ride-hailing demand forecasting problems into multiple graphs, and apply multiple graph convolutional networks on these correlations to model inter-regional information. To overcome the high nonlinearities and complex patterns in traffic flow forecasting problems, Guo *et al.* [41] propose a model named ASTGCN, which separates traffic flows into three temporal properties and utilizes an attention mechanism to capture dynamic spatial-temporal correlations. And they also conduct spatial-temporal convolution on traffic flow data to dig temporal features. Besides, a weighted fusion of different temporal properties is used to generate prediction results. Similar thoughts could be seen in LSGCN [42], which integrates a graph attention network and graph convolution networks into a spatial gated block to accomplish long and short-term prediction tasks. Li *et al.* [43] design novel hyper-networks named DGCRN to discover the dynamic pattern from node attributes with time-variant dynamic filters, which is the first attempt to model the topology of the dynamic graph in a generative way.

Compared with previous work, our model possesses several key differences: i) a joint static-dynamic representation learning network is proposed to capture the patterns of the temporally evolving process and spatially mutual influences of traffic anomalies; ii) a creative evolution strategy is designed for emphasizing the rare anomaly occurrence; iii) a novel loss function is applied to solve the imbalanced data distribution problem.

3 Problem Definition

In this section, we first introduce some important definitions, followed by the formal problem statement.

Definition 1 (**Traffic Anomalies**). Traffic anomalies are the accident events that would disturb the normal traffic pattern and lead to dangerous traffic conditions. In this paper, the traffic anomalies are narrowly defined as traffic collision events. In other words, one traffic anomaly means that there is at least one traffic collision event in a certain area during the time interval. And we define no traffic collision as a normal traffic event.

Definition 2 (**City Segmentation**). A city is partitioned equally into $i \times j$ grids according to latitude and longitude. We observe that motor vehicle collisions rarely occur in some grids, such as parks and lakes. Thus, we ignore such invalid information and mainly focus on the $N \leq i \times j$ grids where collisions may happen. In this way, the traffic features can be extracted with different resolutions by changing i or j based on the requirement.

Definition 3 (**Static Embedding**). Static embedding is the direct encoding of current traffic features, such as POIs, weather conditions and statistical traffic flow, etc., which describes the stationary properties of current traffic events and locations acquired by sensors or other installations directly.

Definition 4 (**Dynamic Embedding**). Dynamic embedding is proposed to encode the time-varying properties and temporal dynamical mutual influence among the traffic events,

which aggregates the current static embedding (**Update**) and evolves over time (**Decay**). It reflects anomaly frequency and capture mutually recursive correlation and dependency between the temporal-adjacency traffic events.

Problem Definition. Given a city with N grids and available features, *grid origin feature representation* is denoted by $O_t \in \mathbb{R}^{N\times D}$ and *grid difference feature representation* is represented by $\mathcal{D}_t \in \mathbb{R}^{N\times D}$, which are elaborated in Sec. 4.1.3. We transform them into static embedding $X_t^s \in \mathbb{R}^{N\times 2D}$ by concatenating O_t with D_t, and learn the dynamic embedding $X_t^d \in \mathbb{R}^{N\times 2D}$ based on the temporal evolution model, where D is the number of adopted features. Afterwards, we concatenate them to obtain $X_t \in \mathbb{R}^{N\times 4D}$. Then, the prediction goal is defined as

$$\hat{Y}_{t+1} = \mathrm{F}([X_{t-T+1}, ..., X_{t-1}, X_t]), \tag{1}$$

where N is the number of city grids, and $4D$ is the dimension of grid features, T denotes the length of historical spatio-temporal series, F is the prediction model, and $\hat{Y}_{t+1} \in \mathbb{R}^N$ is the prediction results within each region and during next certain time interval, *i.e.*, whether there are traffic anomalies in the future.

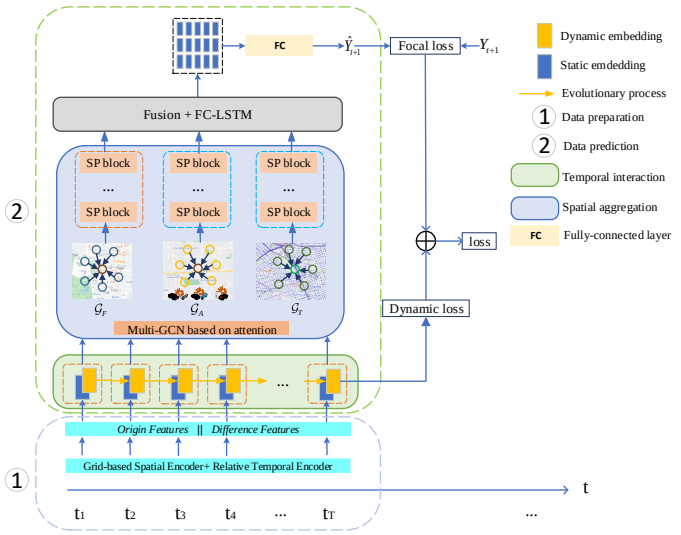


Fig. 2. Overview of the designed deep neural network structure for SNIPER to predict the traffic anomalies. We take dynamic evolution features with static embeddings into account, and a dynamical evolving method and a multi-GCN based on attentions are designed to capture spatio-temporal information. Moreover, a novel loss function combining focal loss and dynamic loss is adopted to tackle the issue of imbalanced data.

4 METHODOLOGY

To predict the traffic anomalies based on multiple publicly available data, we propose a joint static-dynamic spatio-temporal representation learning model called SNIPER. The main deep learning network structure of SNIPER is depicted in Fig. 2. After embedding the data into static representations and dynamic representations based on the encoded spatio-temporal information, an evolving strategy is designed to formulate the temporal dynamic process for the events. Then, multiple graph convolutional networks with attention mechanisms are used to formulate the mutual spatial influence among the events from different perspectives, such as POIs, historical anomalies, and traffic flow, etc. The FC-LSTM combined with a fully connected layer is adopted to transform the fused representations from different graphs into the prediction results. And a loss function consisting of a dynamic loss function and an improved focal loss function is proposed to solve the imbalanced data problem.

4.1 Spatio-temporal Information Encoding

Spatio-temporal information is the key factor to depict the interactions among traffic anomalies, whose mutual influence is nonlinearly changing with the variance of time and distance. To capture the dynamic patterns for these interactions, we design a grid-based spatial encoder and a relative temporal encoder, respectively.

4.1.1 Grid-based Spatial Encoder

One city is partitioned into N grids and one grid's positional information is denoted by the latitude coordinate l_{lat} and longitude coordinate l_{lon} of its center. Inspired by the positional encoding method [44] which distinguishes simple sequential orders of tokens in the sentence, we design a two-dimensional position encoding method to represent l_{lat} and l_{lon}. Taking l_{lat} as an example, it is encoded into a $1\times D$ vector $\{PE(l_{lat},0), PE(l_{lat},1), \ldots, PE(l_{lat},2k), PE(l_{lat},2k+1), \ldots, PE(l_{lat},D-1)\}$. In detail, its position encoding can be calculated using sine and cosine functions based on the odd-even sequence as

$$\begin{aligned} PE\left(l_{lat},2k\right) &= \sin\left(l_{lat}/10000^{2k/D}\right), \\ PE\left(l_{lat},2k+1\right) &= \cos\left(l_{lat}/10000^{2k/D}\right). \end{aligned} \tag{2}$$

Finally, the encoding vector of the grid (l_{lat}, l_{lon}) is obtained by concatenation as

$$PE(l_{lat}, l_{lon}) = PE(l_{lat}) \parallel PE(l_{lon}), \tag{3}$$

where $PE \in \mathbb{R}^{N\times 2D}$ denotes the position encoding matrix of grids.

After encoding, given any fixed differences Δ and Δ' for latitude l_{lat} and longitude l_{lon}, $PE(l_{lat}+\Delta, l_{lon}+\Delta')$ can be represented as a linear function of $PE(l_{lat}, l_{lon})$ [44], so that each location owns unique and interrelated spatial encoding.

4.1.2 Relative Temporal Encoder

Temporal information, especially timestamps of traffic anomalies, is significant for spatio-temporal data mining. Relative anomaly occurrence time represents the time difference between the current time t and the beginning time t_1. Inspired by the representation learning on temporal graphs [45], we use a relative temporal encoding method replacing absolute temporal position to capture additional temporal information, which could indicate the dynamic inner patterns among the events. The temporal encoding function is defined as

$$\begin{aligned} ZE(t) = [&\cos(\omega_1(t-t_1)), \sin(\omega_1(t-t_1)), \ldots, \\ &\cos(\omega_k(t-t_1)), \sin(\omega_k(t-t_1)), \ldots, \\ &\cos(\omega_D(t-t_1)), \sin(\omega_D(t-t_1))], \end{aligned} \tag{4}$$

where $ZE(t) \in \mathbb{R}^{2D}$ denotes temporal encoding of time t, and $\omega_k = 1/10000^{2k/D}$. For any fixed δ, the distance between $ZE(t+\delta)$ and $ZE(t)$ measured by the dot-product transformation [46] can be written as

$$ZE(t+\delta) \cdot ZE(t) = \cos(\omega_1\delta) + \cdots + \cos(\omega_k\delta) + \cdots + \cos(\omega_D\delta), \quad (5)$$

where $\cdot$ denotes dot-product, and the distance which is dependent on δ is also fixed. In this way, every timestamp t has its unique relative encoding and could be used to discover the dynamic patterns in traffic anomalies.

4.1.3 Multi-source Feature Fusion for Static Embedding

Let $O_t \in \mathbb{R}^{N\times D}$ denote the *grid origin feature representation* at timestamp t, including POI, weather condition, and traffic flow, etc. Given the historical observations of each grid, we obtain *grid difference feature representation* $\mathcal{D}_t \in \mathbb{R}^{N\times D}$ by calculating the difference of *grid origin features* between the current sample and the average of the last n normal samples $O^l_{(t_i,\text{nor})}$, which is defined as

$$\mathcal{D}^l_t = O^l_t - \frac{1}{n}\sum\nolimits_{i=1}^{n} O^l_{(t_i,\text{nor})}. \quad (6)$$

Grid difference feature representation $\mathcal{D}^l_t \in \mathbb{R}^D$ describes how current sample deviates the normal samples at grid l, which means anomaly tends to have larger $\mathcal{D}^l_t$ than normal events.

To obtain static embeddings, the spatio-temporal information encoding is fused with aggregated grid features as

$$X^s_{(t,l)} = \left[O^l_t || \mathcal{D}^l_t\right] + PE(l_{lat}, l_{lon}) + ZE(t), \quad (7)$$

where $||$ denotes the concatenation operation, $O^l_t \in \mathbb{R}^D$, $PE(l_{lat}, l_{lon}) \in \mathbb{R}^{2D}$, $ZE(t) \in \mathbb{R}^{2D}$, $X^s_{(t,l)} \in \mathbb{R}^{2D}$, and $X^s_t \in \mathbb{R}^{N\times 2D}$.

4.2 Temporally Dynamical Evolving Embedding

Dynamic embedding is used to represent the evolving interactions among the event list along the timeline, which updates the current learning representation based on the past information. The preliminary analysis on the real-world datasets reveals an imbalance data distribution, in which the anomaly traffic events account for a low rate. During training, the imbalance distribution would lead to the "over-smoothing" problem for the existing methods [36], which only depend on the interactions among the spatio-temporal adjacent event to learn the event patterns. In detail, the representation algorithm operated on the imbalanced data makes latent embeddings more and more similar in the evolution progress due to very high rate of normal events and ignores the significant information of rare anomaly events, which exacerbates the performance of prediction tasks.

As shown in Fig. 3, to capture the traffic anomaly influence among the imbalanced data, we design an evolution strategy by paying more attention to the past rare traffic anomalies during the whole dynamic prediction process, which consists of **Update** and **Decay**. If the traffic event of grid l is anomalous at time interval t, its dynamic embedding $X^d_{(t,l)} \in \mathbb{R}^{2D}$ can be calculated by **Update** operation as

$$X^d_{(t,l)} = \sigma\left(\left[X^d_{(t-1,l)} || X^s_{(t,l)}\right] W_1\right), \quad (8)$$

where σ denotes a sigmoid activation function, $X^s_{(t,l)} \in \mathbb{R}^{2D}$ denotes the static embedding of grid l at time t, $W_1 \in \mathbb{R}^{4D\times 2D}$ is the trainable weight matrix to obtain the dynamic embedding of location l while capturing the influence of recent traffic anomalies. In this way, information about current static anomaly embedding could be fused into dynamic embedding, which could fully capture the low-rate traffic anomaly event patterns. Distinguished from **Update** where the embedding gets updated only when traffic anomalies happen [36], if the traffic event of grid l is normal from time t_θ to time t, its dynamic embedding $X^d_{(t,l)} \in \mathbb{R}^{2D}$ can be calculated by **Decay** operation as

$$X^d_{(t,l)} = X^d_{(t_\theta,l)} \exp\left(-\frac{(t-t_\theta)}{\varphi}\right), \quad (9)$$

where φ is a constant to determine the exponential decay rate. It is obvious that as $(t-t_\theta)$ increases, $X^d_{(t,l)}$ gets smaller. And the magnitude of $X^d_{(t,l)}$ can reflect anomaly frequency in grid l and the interval from the last anomaly to the current. **Decay** operation is used to capture mutually recursive dependency and correlation between the temporal-adjacency traffic events.

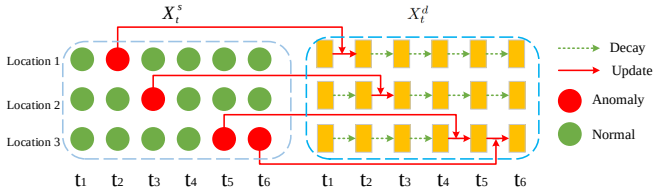


Fig. 3. An example of the evolutionary process for dynamic embeddings, which conducts **Update** operations and **Decay** operations in different locations as time changes. The left are static embeddings of location 1, 2 and 3 from time t_1 to t_6. The right is evolutionary processes of dynamic embeddings for these locations.

An example of the temporal evolutionary operations for the dynamic embedding is illustrated in Fig. 3. Dynamic embeddings of all grids are evolving at the same time, *i.e.*, **Update** and **Decay** occur in different grids simultaneously. When (t_2, l_1), (t_3, l_2), (t_5, l_3) and (t_6, l_3) are anomaly, $(X^d_{(t_1,l_1)}, X^s_{(t_2,l_1)}) \rightarrow X^d_{(t_2,l_1)}$, $(X^d_{(t_2,l_2)}, X^s_{(t_3,l_2)}) \rightarrow X^d_{(t_3,l_2)}$, $(X^d_{(t_4,l_3)}, X^s_{(t_5,l_3)}) \rightarrow X^d_{(t_5,l_3)}$ and $(X^d_{(t_5,l_3)}, X^s_{(t_6,l_3)}) \rightarrow X^d_{(t_6,l_3)}$ conduct **Update** operations as shown in red solid arrow, while other dynamic embeddings conduct **Decay** operations as shown in green dotted arrow. In this way, the proposed method SNIPER could pay more attention on the anomaly events to deal with the imbalanced data distribution, and reduce the negative impact of normal event data accumulation on temporally dynamical evolving embedding learning.

4.3 Spatially Mutual Influence Representation Learning

Inspired by multiple graphs on capturing different types of spatial correlations among the partitioned grids [4], [40], we

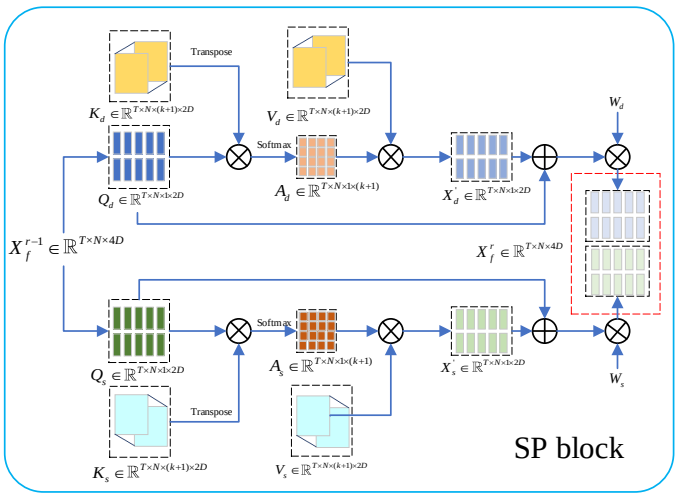


Fig. 4. Overview of the designed attention mechanism for SNIPER. A novel spatial attention mechanism (SP block) using static and dynamic embeddings is proposed to capture the weights of neighbor grids.

construct our multiple graphs from the views of grid function, collision records, and traffic condition. To model these spatial interactions, we propose a multi-graph convolutional network with attention mechanisms to capture the weights of neighbor girds. The module of spatially mutual influence representation learning is shown in Fig. 2.

4.3.1 Multiple Spatial Graph Construction

To model the similarities among grids from different perspectives, we construct three spatial graphs considering various kinds of factors, including the grid functionality graph $\mathcal{G}_F = (V, \mathbf{A}_F)$ which represents the functional similarity of surrounding POIs of different grids, the collision-associated graph $\mathcal{G}_A = (V, \mathbf{A}_A)$ which represents the similarity of past collision records of different grids, such as total collision number, collision reason, and casualties, etc., and the traffic condition graph $\mathcal{G}_T = (V, \mathbf{A}_T)$ which represents the similarity of traffic facility distribution containing various highway types and numbers, and bus stops, etc. In these graphs, $v \in V$ is one vertex denoting one grid, $A_* \in \{A_F, A_A, A_T\}$ is the adjacency matrix of all grids.

The vertex representation of three spatial graphs aggregates its neighbors' information, which contributes to predicting traffic anomalies in a grid. Taking the grid functionality graph $\mathcal{G}_F$ as an example, POI distribution vectors of all grids are normalized by Max-Min normalization [47]. The similarity weight for one edge is obtained by calculating the Euclidean distance of vectors of two vertices. Top-k closest neighbors are chosen to aggregate based on the edge weight, which is denoted by N_k. Hence, the adjacency matrix $A_F(i,j)$ between two vertices (grids) v_i and v_j is defined as

$$A_F(i,j) = \begin{cases} 1, v_j \in N_k(v_i) \cup v_i, \\ 0, \text{otherwise.} \end{cases} \tag{10}$$

4.3.2 Multi-GCN with Attention Mechanisms

We select neighbors for each grid based on three spatial graphs, respectively. In section 4.1 and 4.2, static embedding and dynamic embedding are obtained from two different channels. Distinguished from these existing multi-GCN models [4] [40], we exploit the attention mechanism to learn weights of neighbors from static and dynamic perspectives on each graph rather than feeding them into graph convolutional network directly to formulate the spatial influences. A novel spatial attention mechanism block (SP block) as described in Fig. 4 is proposed in our model to capture the weights of neighbor girds. To present the basic idea of the spatial attention mechanism clearly, three terms are introduced: Query, Key, and Value [44]. It is worth noting that, each graph has its weight matrix. As for each graph $\mathcal{G}_* \in \{\mathcal{G}_F, \mathcal{G}_A, \mathcal{G}_T\}$, attention mechanism is calculated as follows. We set $X_s = [X^s_{t-T+1}, ..., X^s_{t-1}, X^s_t] \in \mathbb{R}^{T\times N\times 2D}$ and $X_d = [X^d_{t-T+1}, ..., X^d_{t-1}, X^d_t] \in \mathbb{R}^{T\times N\times 2D}$ to represent historical static and dynamic embedding lists.

As shown in Fig. 4, take X_d as a example, $Q_d = X_d W_{dq}$ denotes the dynamic query, where $W_{dq} \in \mathbb{R}^{2D\times 2D}$ is a weight matrix. As for each grid, there exists a neighbor set with $k+1$ key-value pairs obtained from equation (10). $X_d(k+1) \in \mathbb{R}^{T\times N\times (k+1)\times 2D}$ denotes the $k+1$ adjacency embeddings of X_d looking up $\mathcal{G}_*$. $K_d = X_d(k+1)W_{dk}$ is the key of Q_d, where $W_{dk} \in \mathbb{R}^{2D\times 2D}$ is a weight matrix. V_d denotes the value, and is equal to K_d, which means the value is obtained from the same source as the key K_d. We can obtain the weights by calculating the dot product between the Q_d and K_d, and applying a softmax function. To be more specific, the dynamic embedding aggregated by the neighbors can be written as

$$X^{'}_d = A_d V_d, \tag{11}$$

where $A_d \in \mathbb{R}^{T\times N\times 1\times (k+1)} = soft\max(\frac{Q_d {K_d}^T}{\sqrt{2D}})$ denotes attention matrix, $X^{'}_d \in \mathbb{R}^{T\times N\times 2D}$ after removing the dimension of size 1. $A_s \in \mathbb{R}^{T\times N\times 1\times (k+1)}$ and $X^{'}_s \in \mathbb{R}^{T\times N\times 2D}$ are obtained in the same way. To optimize training efficiency, inspired by residual learning [48] , the fusion of the static and dynamic embedding after one SP block can be written as

$$X^1_f = \sigma\left(\left[(Q_d + X^{'}_d)W_d\right] \,||\, \left[(Q_s + X^{'}_s)W_s\right]\right), \tag{12}$$

where $X^1_f \in \mathbb{R}^{T\times N\times 4D}$, $W_d \in \mathbb{R}^{2D\times 2D}$ and $W_s \in \mathbb{R}^{2D\times 2D}$ denote kernels with $2D$ filters in the graph convolutional operations from dynamic and static factors, respectively. After r SP blocks, the output can be written as X^r_f that can concatenate neighbor information from r-orders.

4.3.3 Joint Representation Learning for Multi-GCN

Instead of concatenation, we project various graph information into an implicit common space with their respective weight matrices by joint learning. And the fusion can be written as

$$\begin{aligned} \widehat{X}_f = \sigma(X^r_{(f,\mathcal{G}_F)} \odot W_{\mathcal{G}_F} + X^r_{(f,\mathcal{G}_A)} \odot W_{\mathcal{G}_A} \\ + X^r_{(f,\mathcal{G}_T)} \odot W_{\mathcal{G}_T}), \end{aligned} \tag{13}$$

where $\widehat{X}_f \in \mathbb{R}^{T\times N\times 4D}$ is the joint learning representation, $\odot$ is the row-wise product, $W_{\mathcal{G}F}$, $W_{\mathcal{G}A}$, and $W_{\mathcal{G}T} \in \mathbb{R}^{4D}$ are weight matrices acting on feature dimension to reflect the influence levels of the three spatial graphs on the process of predicting traffic anomalies.

One **FC-LSTM** layer [17] which performs well in modeling the dependencies in the spatio-temporal dimensions

combining with a fully connected layer, is applied to aggregate the features from the perspectives of T time steps, $4D$ dimensions and N grids. More specifically, the FC layer of FC-LSTM is used to aggregate the $4D$ channels of each grid to map all features into non-linear representation and fully capture the correlation between all features. The hidden state of FC-LSTM cell at time step t which aggregates feature tensors of T time steps is fed into a fully-connected layer (F) with a sigmoid activation function to transform the output of the FC-LSTM into the expected prediction and capture dependencies among N grids. Then the outputs can be written as

$$\widehat{Y}_{t+1} = \mathrm{F}\left(\textbf{FC-LSTM}\left(\widehat{X}_f\right)\right), \tag{14}$$

where $\widehat{Y}_{t+1} \in \mathbb{R}^N$ denotes the prediction probability of traffic anomalies of N grids.

4.4 Loss Function

To tackle the problem of imbalanced data distribution, we also design an improved focal loss. The label $y = 1$ (positive) and $y = 0$ (negative) correspond to anomaly and normal traffic events, respectively. Then, the focal loss [49] can be written as

$$Floss = -\alpha(1-\widehat{y})^{\gamma} y \log \widehat{y} - (1-\alpha)\widehat{y}^{\gamma}(1-y)\log(1-\widehat{y}), \tag{15}$$

where α and γ are user-defined hyperparameters which are used to control the weight of imbalanced samples. Here, we propose an improved focal loss with a two-classification model to pay more attention to hard-classified samples,

$$\mathcal{L}_F = \begin{cases} 0, \widehat{y}y + (1-\widehat{y})(1-y) > \epsilon, \\ Floss, \text{otherwise}, \end{cases} \tag{16}$$

where ϵ is a probability threshold which is used to pay less attention to well-classified samples with high prediction accuracy. To address the issue of the appropriate change scale of the dynamic characteristics evolution, the dynamic loss is defined as

$$\mathcal{L}_D = \max(0, a - y(t,l)\left|X^d_{(t,l)} - X^d_{(t-1,l)}\right|), \tag{17}$$

where a is a hyperparameter that controls the Euclidean distance of each evolution closer to a fixed threshold to avoid the 'over-smoothing' problem. Obviously, in the **Decay** process, $y(t,l) = 0$, $\mathcal{L}_D = a$ and $\frac{\partial \mathcal{L}_D}{\partial W} = 0$. Thus, there is no gradient on the two temporal-adjacency traffic events, which indicates that $\mathcal{L}_D$ is only effective in **Update** process. In this way, it could pay more attention when anomaly events occur. In the **Update** process, $y(t,l) = 1$, if $\left|X^d_{(t,l)} - X^d_{(t-1,l)}\right| < a$, then $\mathcal{L}_D = a - \left|X^d_{(t,l)} - X^d_{(t-1,l)}\right|$, and $\mathcal{L}_D$ endeavors to distance $X^d_{(t,l)}$ from $X^d_{(t-1,l)}$ to avoid the 'over-smoothing' problem as mentioned in Sec. 4.2. If $\left|X^d_{(t,l)} - X^d_{(t-1,l)}\right| > a$, then $\mathcal{L}_D = 0$ and $\frac{\partial \mathcal{L}_D}{\partial W} = 0$, and $\mathcal{L}_D$ no longer enlarges the mapping distance between $X^d_{(t,l)}$ and $X^d_{(t-1,l)}$ to keep the inherent correlation and dependency between the two temporal-adjacency traffic events.

Based on the analysis above, the final loss function can be written as

$$\mathcal{L} = \mathcal{L}_F + \lambda \mathcal{L}_D, \tag{18}$$

where $\lambda \in (0,1)$ is the weight of $\mathcal{L}_D$.

4.5 Effective Model Training with Evolutionary Strategy

Here we propose a batching algorithm to parallelize the training process of SNIPER, which could improve the training efficiency. The key point of parallelization is to maintain temporal dependencies during the dynamic evolution process when training the model. However, the evolution strategies for all grids are not all the same and change over time. This prevents us from simply splitting time slices into individual batches and processing them in parallel, which is conducted in existing methods [2], [3], [4].

To solve this problem, we take evolution interactions at all grids into consideration in parallel, and train the batches chronologically. To be more specific, instead of training dynamic embedding iteratively, we complete recursion evolution in the model before training of each batch. Our training method works in batch in two steps, *i.e.*, the record step and the recursive step. In the record step, take the k-th batch B_k as an example, its sample list can be represented as $[B^1_k, B^2_k, ...B^S_k]$, where S denotes batch size. As shown in Fig. 3, as for one grid l, the evolution list $L_{(B_k,l)} = [Y_{(t_{(B^1_k)},l)}, Y_{(t_{(B^2_k)},l)}, ..., Y_{(t_{(B^S_k)},l)}]$ depends on the label of the current time slice. In the recursive step, the dynamic embedding of all samples in B_k get evolved by feeding $X^d_{(t_{(B^S_{k-1})},l)}$ into B_k. As a consequence, $X^d_{(t_{(B^1_k)},l)} = E_{Y_{(t_{(B^1_k)},l)}}(X^d_{(t_{(B^S_{k-1})},l)})$, $X^d_{(t_{(B^2_k)},l)} = E_{Y_{(t_{(B^2_k)},l)}}(X^d_{(t_{(B^1_k)},l)})$, $\cdots$, $X^d_{(t_{(B^S_k)},l)} = E_{Y_{(t_{(B^S_k)},l)}}(X^d_{(t_{(B^{S-1}_k)},l)})$ where E denotes evolution function, $E_{Y=1}$ and $E_{Y=0}$ denote **Update** operations and **Decay** operations, respectively. All the dynamic embeddings are evolved and trained together utilizing the dynamic embedding $X^d_{(t_{(B^S_{k-1})},l)}$ and the evolution list of batch B_k in the recursive evolution process. Algorithm 1 outlines the pseudocode of the main training process of SNIPER. We feed training data into our model, and optimize SNIPER by gradient descent until meeting the stopping criteria.

4.6 Time Complexity of SNIPER

For the temporal evolutionary embedding (Sec. 4.2), the computational complexity is $O\left(D^2NT + DNT\right)$ for **Update** operation, and $O\left(DNT\right)$ for **Decay** operation, so the computational complexity in this part is $O\left(D^2NT + DNT\right)$. For the spatial influence aggregation part (Sec. 4.3.2), the computational complexity is $O\left(\left(DNT(k+1) + NT(k+1) + D^2NT\right) rn_g\right)$, where n_g denotes the number of graph types. In the joint learning part (Sec. 4.3.3), the computational complexity of FC-LSTM is $O\left(D^2NT\right)$, and the computational complexity of the last fully-connected layer is $O\left(N^2\right)$. Because $T < (k+1) < N < D$, the total computational complexity of SNIPER approximates to $O\left(D^2NTrn_g\right)$.

5 Experiments

In this section, we evaluate the performance of our model, SNIPER. We introduce the dataset information for evaluation (Sec. 5.1), detail the parameter setting, evaluation metrics and competitors (Sec. 5.2, Sec. 5.3 and Sec. 5.4),

Algorithm 1 Algorithm of SNIPER

Input: Origin feature representation of N grids O_t, Adjacency matrices A_F, A_A and A_T, hyperparameters such as the length of time slices T, the number of SP blocks r and batch size S, labels Y_t

Output: A learned model SNIPER

Calculate $D_{(t,l)}$, $PE(l_{lat}, l_{lon})$ and $ZE(t)$
Obtain the static embedding by equation (7)
Split all data following chronological order
Initialize all trainable parameters of SNIPER
$epoch \leftarrow 0$
while $epoch \leq epoch_{\max}$ **do**
  Initialize dynamic embedding $X^d_{t_{(B_0^S)}}$ to 0
  $k \leftarrow 1$
  while $k \leq k_{\max}$ **do**
    Select batch B_k
    As for n-th sample of batch B_k, $X_s(B_k^n, l) \leftarrow [X^s_{(t_{(B_k^n)}-T+1,l)}, X^s_{(t_{(B_k^n)}-T+2,l)}, \cdots, X^s_{(t_{(B_k^n)},l)}]$
    Calculate $X_d(B_k)$, utilizing $X_s(B_k)$, $X^d_{t_{(B_{k-1}^S)}}$ and L_{B_k} among N grids and T time slices in parallel with evolution strategy
    Calculate $X^r_{(f,\mathcal{G}_F)}(B_k)$, $X^r_{(f,\mathcal{G}_F)}(B_k)$, $X^r_{(f,\mathcal{G}_F)}(B_k)$ by spatially mutual influence aggregation with attention mechanism using A_F, A_A and A_T
    Calculate $\widehat{X}_f(B_k)$ by equation (13)
    $\widehat{Y}_{t+1}(B_k) \leftarrow \mathrm{F}\left(\textbf{FC-LSTM}\left(\widehat{X}_f(B_k)\right)\right)$
    Calculate $\mathcal{L}_F$ and $\mathcal{L}_D$, and $\mathcal{L} \leftarrow \mathcal{L}_F + \lambda\mathcal{L}_D$
    Forward-backward on $\mathcal{L}$ using the training dataset of B_k based on Adam gradient descent with an adjustable learning rate
    $k \leftarrow k + 1$
  end while
  if Early stopping criteria is met on validation set
    break
  else
    $epoch \leftarrow epoch + 1$
  end if
end while
return A trained model SNIPER

present the experimental comparison (Sec. 5.5), show the effects of different settings on the prediction results (Sec. 5.6), verify the efficiency of the evolutionary strategy for training SNIPER (Sec. 5.7), and visualize our prediction results and ground truth on the Google Map to provide the information in an intuitive way (Sec. 5.8).

5.1 Datasets

We conduct extensive experiments on two public real-world datasets consisting of five different kinds of data collected from NYC[3], Chicago[4] and OpenStreetMap[5] to validate our model. The detailed information of the datasets is shown in Table 1. The traffic collision event data includes latitude, longitude, time, and other collision records; The POI data is composed of garage, school, commercial, and supermarket, etc. The traffic facility data includes bus stop, motorway, cycleway, path, and traffic signals, etc. The historical hourly weather data represents weather descriptions such as rain, snow, wind speed, and temperature, etc. The taxi flow data includes the time and location of pickup and dropoff. And the number of *grid origin feature* D is 58 in total. The statistics of anomaly rate for NYC and Chicago traffic event datasets with the different number of grids are shown in Table 2. It reveals that anomaly events usually account for a low proportion in one dataset, and the proportion becomes lower with the segmentation number increasing. In order to show the imbalance data distribution intuitively, we randomly choose one grid from each segmentation with a different grid-scale, and visualize the traffic anomalies in Fig. 5, respectively.

3. https://opendata.cityofnewyork.us/
4. https://data.cityofchicago.org/
5. https://www.openstreetmap.org/

TABLE 1
Detailed Information of Datasets

Dataset	NYC	Chicago
Time range	1/1/2015-6/30/2016	1/1/2016-12/31/2016
Collision events	306.9k	44.2k
POI	1.3m	839.7k
Traffic facility	307.0k	142.7k
Weather	13,128	8,764
Taxi flow	211.7m	27.5m

TABLE 2
Anomaly rate of different city segmentation.

Datasets	Grid scale	Anomaly rate
Chicago	4×4	0.3023
	6×6	0.1442
	8×8	0.0988
	10×10	0.0671
NYC	8×8	0.2522
	10×10	0.2240
	12×12	0.1818
	14×14	0.1456

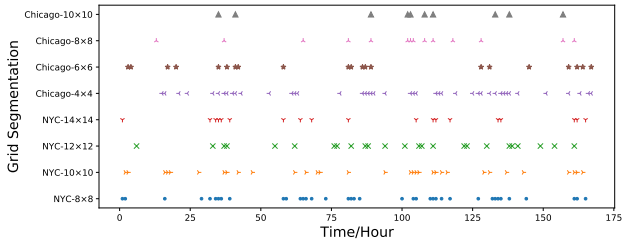


Fig. 5. Illustration of anomaly events with different city segmentation (7 days).

5.2 Parameter Settings

All datasets are split into training sets, validation sets, and test sets by the ratio of 6:2:2 following chronological order. All datasets are normalized by Max-Min normalization into the range [0, 1]. The interval of the time slice is set to 1 hour.

The hyperparameters are determined by our model's performance on the validation datasets. As for the length of historical spatio-temporal series, we set T to 5. For the number of last normal samples, n is set to 5. For the exponential decay constant, φ is set to 2. We set the number of neighbors k to 10 and the number of SP blocks to 6. As for hyperparameters of focal loss, we set $\alpha = 0.25$, $\gamma = 2$, and $\epsilon = 0.9$. The threshold a is set to 6. And the weight λ is set to 0.005. Our batch size S is set to 24. The optimized solutions are searched based on the Adam gradient descent algorithm [50], and the learning rate is set to 0.001.

5.3 Evaluation Metrics

- **AUC-PR**: AUC-PR is an area under the curve of precision against recall. In the context of anomaly prediction, AUC-PR might be a more appropriate metric for evaluation than AUC-ROC. The AUC-ROC value can be affected by the nature of imbalance in anomaly prediction data, in which the normal class takes a large proportion. On the contrary, AUC-PR evaluates precision and recall, which avoids the influence of data imbalance. However, due to the heterogeneous distributions of anomalies, it is challenging for AUC-PR to achieve a high AUC-PR value.
- **AUC-ROC**: By definition, AUC-ROC refers to the area under the ROC curve of true positive rate against false positive rate, which reflects the performance of the model in general. Specifically, the maximum value of AUC-ROC is 1, which indicates the best performance of the model. And 0.5 indicates a random score attributed to objects.
- **F1 score**: F1 score conveys the balance of precision and recall by taking the harmonic mean of these two.
- **Accuracy**: Accuracy measures how many samples, both normal and anomaly, are correctly classified.

5.4 Competitors

We compare our model SNIPER with 5 baseline algorithms and 5 state-of-the-art algorithms consisting of both classical machine learning methods and neural-network-based methods, and we choose optimum parameters for the competitors by their performance on validation sets with grid search method also. Their descriptions and the main parameters are summarized as follows.

5.4.1 Classical machine learning methods

- **HA:** Historical Average, which models the collision events using the average of previous time slices as the prediction, and the last 10 time slices are used to predict the next value.
- **ARIMA [51]:** Auto-Regressive Integrated Moving Average is a well-known model for understanding and predicting events in the type of time series. We train a separate ARIMA model for each grid, and predict traffic anomalies individually, during the experiments.
- **Linear Regression (LR) [11]:** Linear Regression is a state-of-the-art method to predict the possible crime event. It analyzes the historical crime records and models the prediction problem based on a linear regression algorithm. In this problem, we take all of our origin features as the input to the LR model.
- **LightGBM [52]:** LightGBM is one of the most popular models based on the highly efficient boosting decision tree in data mining tasks as well as a state-of-the-art model. We set its main parameters such as max depth, number of leaves, and number of boost round by searching grid over {4, 6, 8, 10}, {20, 40, 80, 140}, and {100, 150, 250, 400}, respectively.
- **XGBoost [53]:** XGBoost is an efficient method based on the tree boosting and used widely in data mining challenges. We set its main parameters such as max depth, min child weight, and max delta step by searching grid over {3, 6, 9, 12}, {1, 2, 4, 8}, and {0, 1, 2, 4}, respectively.
- **CatBoost [54]:** CatBoost is an ensemble technique that endeavors to achieve accurate results from various practical tasks. We set its main parameters such as depth, iterations, and learning rate by searching grid over {4, 8, 16, 32}, {150, 300, 450, 600}, and {0.05, 0.1, 0.15, 0.2}, respectively.

5.4.2 Neural-network-based methods

- **LSTM [55]:** Long Short-Term Memory network is an artificial RNN architecture for sequence learning, which could capture distant dependency in time series. We conduct the grid search on the length of input time series over {2, 4, 6, 8, 10}.
- **ConvLSTM [56]:** ConvLSTM combines CNN and LSTM and performs well in modeling the correlations in the spatio-temporal dimensions. We search the optimum parameters on the length of input time series over {2, 4, 6, 8, 10} and the size of convolution kernel over {1, 2, 4, 8}.
- **STG2Seq [57]:** Spatio-Temporal Graph to Sequence Model is a state-of-art model to formulate the spatio-temporal event. They take advantage of a hierarchical graph convolutional structure to capture the temporal and spatial correlations using historical traffic anomaly numbers. We set its closeness sequence length to {2, 4, 6, 8, 10} and the spatial neighbors number to {2, 4, 8, 16}.
- **GSnet [4]:** GSnet takes geographical and semantic aspects into account to learn spatio-temporal correlations among regions and utilizes multi-source spatio-temporal factors to forecast traffic accident risk, which is also a state-of-art model. We conduct grid search on the geographical convolution kernel size over {1, 2, 4, 8}, and the number of semantic neighbors over {2, 4, 8, 16}.

5.5 Performance Comparison

We compare our model with above eight baselines on New York City and Chicago datasets. We split New York City map into 10 × 10 grids, with grid size 4.6km × 5.4km, and Chicago map into 6 × 6 grids, with grid size 7.0km × 5.6km.

The specific prediction performance comparison of different approaches is shown in Table 3. For performance on Chicago dataset, SNIPER improves the state-of-the-art

TABLE 3
Performance comparison on the NYC and Chicago traffic datasets. The best performances are in bold and the second-bests are marked by $*$.

Method		HA	ARIMA	LR	XGBoost	CatBoost	LightGBM	LSTM	ConvLSTM	STG2Seq	GSnet	SNIPER
Dataset	Metric											
NYC	AUC-PR	0.5827	0.2394	0.6102	0.6201	0.6134	**0.6341**	0.5899	0.6078	0.5083	0.5890	0.6264*
	AUC-ROC	0.8277	0.5068	0.8372	0.8417	0.8365	0.8490*	0.7697	0.8410	0.7640	0.8469	**0.8507**
	F1 score	0.6103	0.2875	0.6079	0.6116	0.6109	0.6141	0.6155*	0.5967	0.5650	0.6128	**0.6262**
	Accuracy	0.7944	0.5954	0.8081	0.8143	0.8097	0.8204	0.8068	0.8057	0.7959	0.8291*	**0.8366**
Chicago	AUC-PR	0.4680	0.1442	0.4352	0.4673*	0.4605	0.4458	0.4046	0.4616	0.3893	0.4662	**0.4704**
	AUC-ROC	0.7540	0.4994	0.7786	0.7804	0.7792	**0.7894**	0.7676	0.7450	0.6657	0.7809	0.7829*
	F1-score	0.4064	0.2240	0.4004	0.4023	0.3971	0.4059*	0.3782	0.3974	0.3710	0.3872	**0.4137**
	Accuracy	0.8382	0.7833	0.8770	0.8792*	0.8697	0.8617	0.8668	0.8704	0.8499	0.8657	**0.8912**

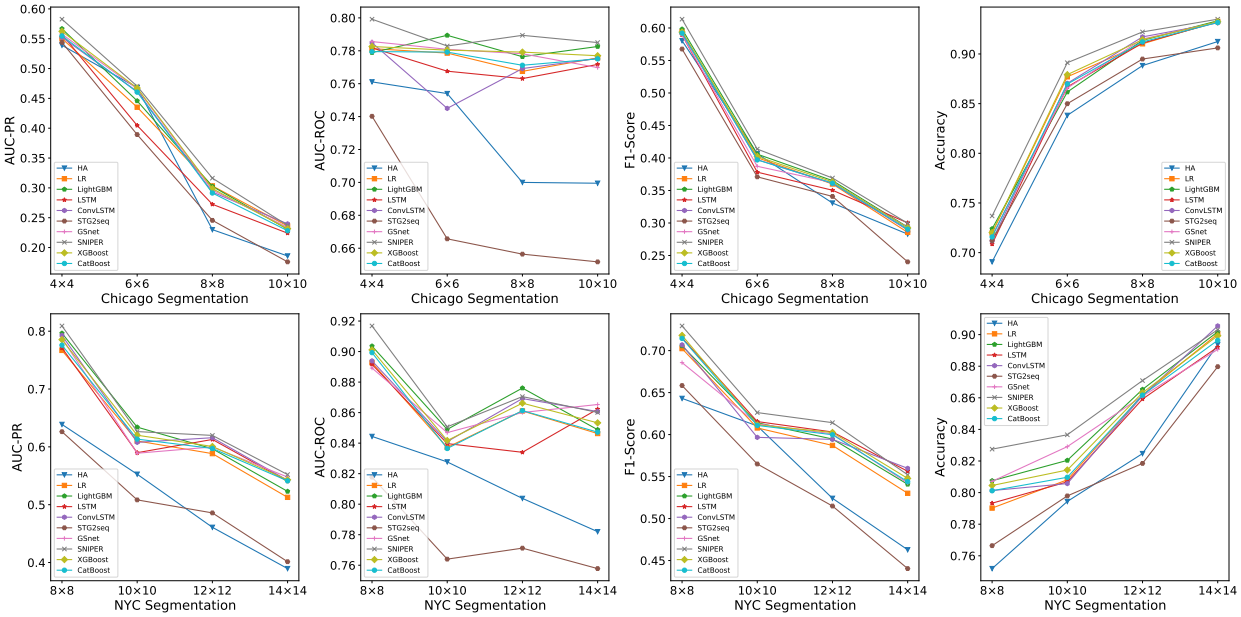


Fig. 6. Four types of grid segmentation in Chicago and NYC. Our SNIPER consistently surpasses other baseline methods for most cases.

method LightGBM which performs the second-best by 5.5%, 1.9%, 3.4% in terms of AUC-PR, F1 score, and accuracy, respectively, except for AUC-ROC which is slightly smaller for regulating numerous parameters to lower errors. In NYC dataset results, SNIPER improves the state-of-the-art method LightGBM by 0.2%, 1.9%, 2.0% in terms of AUC-ROC, F1 score, and accuracy, except for AUC-PR for little improvement. Table 3 shows that our SNIPER consistently outperforms other baseline methods on NYC and Chicago datasets in most cases.

Furthermore, Fig. 6 shows the effect of changing grid size on prediction performance for different methods, where we split Chicago city into 4×4, 6×6, 8×8, 10×10 grids, and split NYC into 6×6, 8×8, 12×12, 14×14 grids, respectively. By thoroughly observing the results, we find ARIMA performs worst in all cases. Thus, the results of ARIMA are removed to amplify the difference of our SNIPER and the other baselines in Fig. 6. The average performance of SNIPER on four segmentation types surpasses the-state-of-the-art method (LightGBM) with the best performance by 3.9%, 0.9%, 1.9% and 1.6% on Chicago datasets, and by 2.4%, 0.6%, 2.6% and 1.3% on New York City datasets in terms of AUC-PR, AUC-ROC, F1 score, and accuracy. In general, as the number of the split grid becomes larger, the positive rate becomes smaller and the prediction task becomes more difficult. As a result, the accuracy metric shows a decelerating trend for all methods and the prediction performances also drop for other metrics. Meanwhile, Fig. 6 shows that our SNIPER delivers the best performance among four segmentation types over Chicago and NYC, respectively.

ARIMA and HA do not perform well due to the fact that they do not consider the information from the spatio-temporal dimension. STG2Seq takes spatio-temporal graphs into consideration but cannot perform well in the prediction tasks. And the limitation of STG2Seq is that it overlooks the imbalanced data distribution and the complex process of traffic anomalies. LR and LSTM can capture the correlations of temporal series data but cannot effectively utilize the dependencies among spatial grids. XGBoost and CatBoost are strong parameter-driven models and cannot tackle the imbalanced data effectively by paying more attention to sparse traffic anomaly events. ConvLSTM combines the advantages of CNN and LSTM to capture significant information from spatial and temporal aspects. But it fails

to aggregate the useful information from each grid's neighbors. GSnet utilizes multiple GCNs to obtain the correlations between grids in spatio-temporal dimension and a temporal attention mechanism to obtain different influences of neighbors, which contribute to obtaining some good results. But the ignored complex dynamic process in this prediction problem includes the sparse but significant inherent information for traffic anomaly events. It can be easily observed that LightGBM performs the second-best. This is mainly because it is a powerful boosting decision tree model which could consider temporal data mining, but it is also hard for LightGBM to improve prediction performance as grid size changes for the lack of attention to imbalanced data.

Overall, although the classical machine learning methods achieve some good results, it does not mean that deep learning-based models are not needed for this anomaly prediction task. First of all, elaborate design of feature engineering is needed for traditional machine learning models, which limits the possibility for applying them on different datasets. On the other hand, deep learning models are able to learn underlying representations of plain data adaptively. Such mechanism makes deep learning model perform better in terms of transferablity and adaption on different datasets than traditional machine learning models like LR and LightGBM. Moreover, in a traffic anomaly prediction task where only a few anomaly data exists, it is hard to capture inherent patterns about traffic anomalies. Traditional machine learning models like LR and LightGBM usually take fixed gird features as input. During the process of training, they try to come up with simple classification strategies for current data, which could be learned more easily than the deep learning-based method based on the imbalanced distribution data. However, these data-driven methods cannot surpass among all methods in all settings for lack of fully considering the spatio-temporal dependency information. Existing classical deep learning methods like LSTM and ConvLSTM are designed to capture the time dependency in time series. However, the extreme sparsity of anomaly data distribution limits the information that deep learning-based models with more parameters could get from time dependency. Furthermore, they do not pay extra attention to the dynamic anomaly patterns, which further explains the poor results they get on traffic anomaly prediction tasks. Thus, in traffic anomaly prediction tasks, traditional temporal-dependency-aware deep learning-based methods perform worse than result-oriented machine learning methods.

Faced with these challenges, our SNIPER effectively combines static embeddings with dynamic embeddings to capture the spatio-temporal correlations and interactive information between locations and traffic events. Our novel attention mechanism and the designed loss function leverage the joint static-dynamic embedding model in the dynamic evolution process to pay more attention to the rare anomaly events. In addition, SNIPER takes the different influences of each neighbor into account, and utilizes spatial attention blocks to obtain the weight of all grid neighbors at each time slice. These novel ideas contribute to our deep learning-based method SNIPER overcoming the above challenges and achieving the best performances.

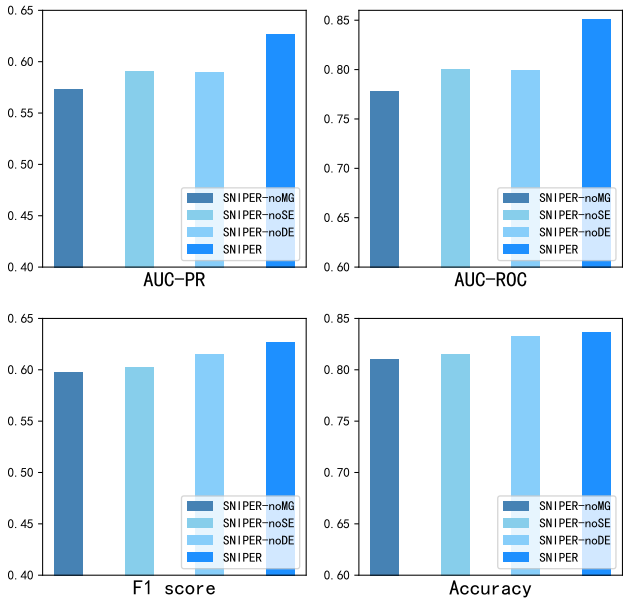


Fig. 7. Components analysis of different network configurations.

5.6 Study Performance of Different Components

To further understand the influence of different components in SNIPER, we conduct three ablation experiments and analyze the experimental performance on NYC datasets.

5.6.1 Ablation experiments of different network configurations

To further investigate the effect of different modules of SNIPER, we design 3 variants of the SNIPER network and compare these 3 variants with SNIPER model. The different structures of these models are illustrated as below:

- **SNIPER-noMG**: We remove the multi-GCN based on attention module to study the contribution of multi-GCN when aggregating the information of neighbors.
- **SNIPER-noSE**: We remove the static embedding module to evaluate the effect of static features when predicting the traffic anomalies.
- **SNIPER-noDE**: We remove the dynamic embedding module to illustrate the performance of dynamic features in the dynamic interactive process.

Fig. 7 gives the detailed prediction results of these models. We can observe that SNIPER-noMG performs the worst, demonstrating the importance of neighbors' information aggregation. It also illustrates that spatial dependency is significant in anomaly prediction tasks. SNIPER-noSE has poorer performance than SNIPER-noDE, which demonstrates that static features play a more vital role than dynamic interactive features in the model. It is obvious that the results of SNIPER-noSE and SNIPER-noDE are worse than SNIPER model, which shows that both static embedding and dynamic embedding are crucial for anomaly data mining.

TABLE 4
Comparison of different loss functions.

Loss Function	MAE	Weighted Cross Entropy	Ours
AUC-PR	0.5874	**0.6272**	0.6264
AUC-ROC	0.7718	0.8424	**0.8507**
F1 score	0.6049	0.6145	**0.6262**
Accuracy	0.8149	0.8254	**0.8366**

5.6.2 Component analysis of loss function

Weighted Cross Entropy (WCE) [58] performs well in classifying imbalanced data, and it can be written as follows

$$WCE = -(1 - r_p)y \log \widehat{y} - r_p(1 - y) \log(1 - \widehat{y}), \quad (19)$$

where r_p denotes the positive rate in training sets. We compare our loss function with MAE (Mean Absolute Error) and Weighted Cross Entropy. Table 4 shows that MAE performs worst for the lack of consideration about imbalance data problem. Our loss function surpasses Weighted Cross Entropy by 0.9%, 1.9% and 1.3% in terms of AUC-ROC, F1 score and accuracy, except for AUC-PR which is slightly smaller. The comparison further demonstrates that our loss function is useful for imbalanced data learning.

5.6.3 Contributions of three constructed spatial graphs

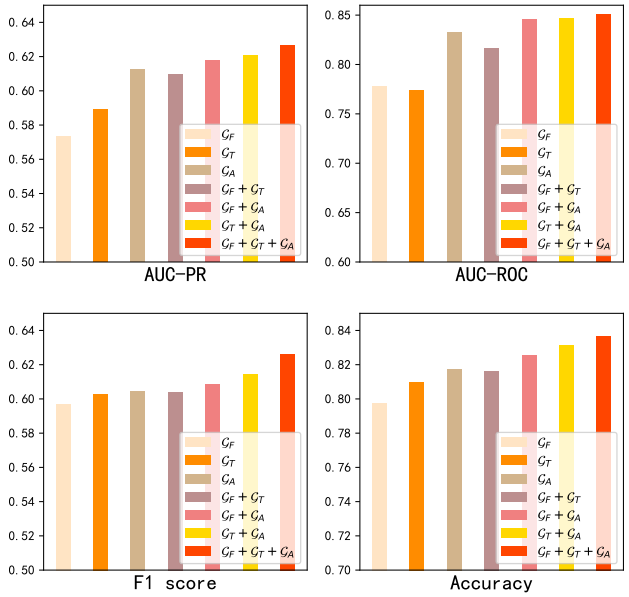


Fig. 8. Component analysis of three constructed graphs $\mathcal{G}_F$, $\mathcal{G}_T$ and $\mathcal{G}_A$ indicating the mutually spatial influences.

We also evaluate the contributions of grid functionality graph $\mathcal{G}_F$, collision-associated graph $\mathcal{G}_T$ and traffic condition graph $\mathcal{G}_A$ as well as their combinations with each other. Thus, there are 6 variants which are constructed to be compared with ($\mathcal{G}_F + \mathcal{G}_T + \mathcal{G}_A$), i.e, SNIPER. Fig. 8 shows that $\mathcal{G}_F$ performs worst among three graphs. $\mathcal{G}_F$ represents the POI similarity of different grids which owns limited traffic information and has a weaker correlation with the predicted traffic anomalies. Compared with $\mathcal{G}_F$ and $\mathcal{G}_T$, $\mathcal{G}_A$ contributes the most to model performance. $\mathcal{G}_A$ represents the similarity of past collision records. It can reflect the general pattern of traffic accident distribution among all grids. Thus, it is more suitable to choose relevant neighbors than $\mathcal{G}_F$ and $\mathcal{G}_T$ even the combination of the other two. All in all, the combination of those three graphs fuse useful information from different perspectives and greatly boost the performance of our model.

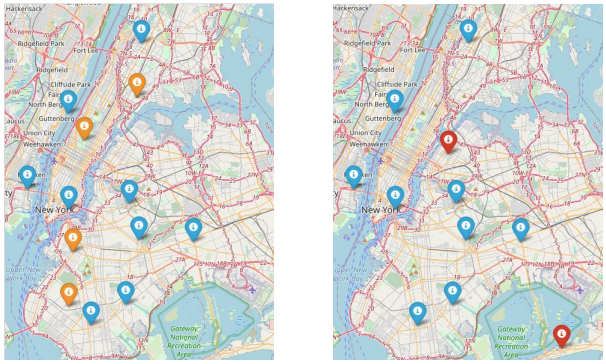
5.7 Training Efficiency

TABLE 5
Comparison of running time every epoch.

Dataset	NYC	Chicago
Runtime of training singly	327.26s	58.32s
Runtime of training in batch	**153.29s**	**44.38s**

Here we compare the running time of SNIPER when training in batch and singly. Table 5 shows the comparison of runtime (in seconds) of each epoch (training on one NVIDIA TITAN V GPU device) with our batch size set to 24. We find that our batching method results in $2.13\times$ and $1.31\times$ speed-up of model training on NYC datasets and Chicago datasets, respectively. This shows that SNIPER is faster and able to train the recursive model as non-recursive models, because of our training batching algorithm using in the model.

5.8 Visualization



(a) Markers of prediction (b) Markers of ground truth

Fig. 9. Markers of prediction results and ground truth. Blue markers denote grids where traffic anomalies occur but they are predicted by our SNIPER. Yellow markers denote false alarms, and red markers denote missing alarms.

To present our prediction results more intuitively, we choose a time slice (2016-06-01 16:00 - 2016-06-01 17:00) randomly and mark them and ground truth data in Google Maps on the NYC dataset with 10×10 grids, which is shown in Fig. 9. There are 11 grids where traffic anomalies occur in total as shown in Fig. 9(b). We have predicted 9 grids of them and 2 grids are missing alarms. Meanwhile, 4 grids are false alarms as shown in Fig. 9(a). The results indicate that our model achieves a reasonably good predicting performance. However, there are still some wrong prediction cases at several grids because of the variability and complexity of predicted traffic anomalies.

6 Conclusion and Future Work

In this paper, we propose a novel traffic anomaly prediction model SNIPER based on joint static-dynamic spatio-temporal evolutionary learning. To indicate the dynamical patterns in the spatio-temporal events, we design a grid-based spatial encoder and a relative temporal encoder. To formulate the interactions among the events along the timeline, we propose a temporally dynamical evolving embedding method. To capture the spatially mutual influence among the partitioned grids from different perspectives, we propose a multi-GCN model with attention mechanisms. To tackle the imbalanced data problem, we design a loss function combining the dynamic loss function and improved focal loss function. To verify the performance of SNIPER, extensive experiments are implemented on two real-world traffic collision-related datasets to show its effectiveness.

In future work, we will further study the effectiveness and robustness of the joint static-dynamic spatio-temporal representation learning model. For example, how to conditionally select the most contributed features among the massive datasets should be interesting, which could significantly improve the robustness of the model. It would also be exciting to explore the feature aggregation strategy in an unsupervised way to improve the prediction effectiveness.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406900, National Natural Science Foundation of China under Grant 61902308, 61822309, 61773310, U1736205, U1766215, Initiative Postdocs Supporting Program BX20190275, BX20200270, and China Postdoctoral Science Foundation 2019M663723, 2021M692565, and the Fundamental Research Funds for the Central Universities under grant xjh032021058, xxj022019016, xtr022019002.

References

[1] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[2] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 3, pp. 468–478, 2019.

[3] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[4] B. Wang, Y. Lin, S. Guo, and H. Wan, "Gsnet: Learning spatial-temporal correlations from geographical and semantic aspects for traffic accident risk forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4402–4409.

[5] D. Lord, S. D. Guikema, and S. R. Geedipally, "Application of the conway–maxwell–poisson generalized linear model for analyzing motor vehicle crashes," *Accident Analysis & Prevention*, vol. 40, no. 3, pp. 1123–1134, 2008.

[6] K. El-Basyouny and T. Sayed, "Collision prediction models using multivariate poisson-lognormal regression," *Accident Analysis & Prevention*, vol. 41, no. 4, pp. 820–828, 2009.

[7] C. Dong, D. B. Clarke, X. Yan, A. Khattak, and B. Huang, "Multivariate random-parameters zero-inflated negative binomial regression model: An application to estimate crash frequencies at intersections," *Accident Analysis & Prevention*, vol. 70, pp. 320–329, 2014.

[8] M.-I. M. Imprialou, M. Quddus, D. E. Pitfield, and D. Lord, "Re-visiting crash–speed relationships: A new perspective in crash modelling," *Accident Analysis & Prevention*, vol. 86, pp. 173–185, 2016.

[9] D. Agarwal, B.-C. Chen, and P. Elango, "Spatio-temporal models for estimating click-through rate," in *Proceedings of the 18th International Conference on World Wide Web*, 2009, pp. 21–30.

[10] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2011, pp. 1010–1018.

[11] H. Wang, D. Kifer, C. Graif, and Z. Li, "Crime rate inference with big data," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2016, pp. 635–644.

[12] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1720–1730.

[13] X. Liu, C. Shen, Y. Fan, X. Liu, Y. Zhou, and X. Guan, "A co-evolutionary model for inferring online social network user behaviors," in *2018 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*. IEEE, 2018, pp. 85–90.

[14] X. Liu, C. Shen, W. Wang, and X. Guan, "Coevil: A coevolutionary model for crime inference based on fuzzy rough feature selection," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 806–817, 2019.

[15] A. Reinhart and J. Greenhouse, "Self-exciting point processes with spatial covariates: modelling the dynamics of crime," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 67, no. 5, pp. 1305–1329, 2018.

[16] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda, "Deep mixture point processes: Spatio-temporal event prediction with rich contextual information," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 373–383.

[17] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms," in *International Conference on Machine Learning*. PMLR, 2015, pp. 843–852.

[18] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, 2021.

[19] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in iot using hyperellipsoidal clustering," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1174–1184, 2017.

[20] A. Essien, I. Petrounias, P. Sampaio, and S. Sampaio, "A deep-learning model for urban traffic flow prediction with traffic events mined from twitter," *World Wide Web*, pp. 1–24, 2020.

[21] S. Wang, J. Cao, and P. Yu, "Deep learning for spatio-temporal data mining: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[22] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[23] Q. Wu, C. Yang, H. Zhang, X. Gao, P. Weng, and G. Chen, "Adversarial training model unifying feature driven and point process perspectives for event popularity prediction," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 517–526.

[24] J. Yu, Y. Lee, K. C. Yow, M. Jeon, and W. Pedrycz, "Abnormal event detection and localization via adversarial event prediction," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[25] P. Chapfuwa, C. Tao, C. Li, C. Page, B. Goldstein, L. C. Duke, and R. Henao, "Adversarial time-to-event modeling," in *International Conference on Machine Learning*. PMLR, 2018, pp. 735–744.

[26] D. Ding, M. Zhang, X. Pan, M. Yang, and X. He, "Modeling extreme events in time series prediction," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1114–1122.

[27] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 3009–3017.

[28] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2007, pp. 1–8.
[29] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 353–362.
[30] R. Tan, R. Chirkova, V. Gadepally, and T. G. Mattson, "Enabling query processing across heterogeneous data models: A survey," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 3211–3220.
[31] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering." in *Advances in Neural Information Processing Systems*, vol. 14, no. 14, 2001, pp. 585–591.
[32] S. Cao, W. Lu, and Q. Xu, "Grarep: Learning graph representations with global structural information," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 2015, pp. 891–900.
[33] M. Ou, P. Cui, J. Pei, Z. Zhang, and W. Zhu, "Asymmetric transitivity preserving graph embedding," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2016, pp. 1105–1114.
[34] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "Representation learning over dynamic graphs," *arXiv preprint arXiv:1803.04051*, 2018.
[35] P. Goyal, S. R. Chhetri, and A. Canedo, "dyngraph2vec: Capturing network dynamics using dynamic graph representation learning," *Knowledge-Based Systems*, vol. 187, p. 104816, 2020.
[36] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1269–1278.
[37] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.
[38] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.
[39] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *International Joint Conference on Artificial Intelligence*, 2019, pp. 1907–1913.
[40] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3656–3663.
[41] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.
[42] R. Huang, C. Huang, Y. Liu, G. Dai, and W. Kong, "Lsgcn: Long short-term traffic prediction with graph convolutional networks." in *IJCAI*, 2020, pp. 2355–2361.
[43] F. Li, J. Feng, H. Yan, G. Jin, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *arXiv preprint arXiv:2104.14917*, 2021.
[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.
[45] D. Xu, C. Ruan, E. Korpeoglu, S. Kumar, and K. Achan, "Inductive representation learning on temporal graphs," in *International Conference on Learning Representations*, 2020.
[46] Y. Lin, H. Wan, S. Guo, and Y. Lin, "Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4241–4248.
[47] Z. Zhao, A. Kleinhans, G. Sandhu, I. Patel, and K. Unnikrishnan, "Capsule networks with max-min normalization," *arXiv preprint arXiv:1903.09662*, 2019.
[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
[49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[51] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.
[52] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3146–3154, 2017.
[53] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
[54] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: unbiased boosting with categorical features," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 6639–6649.
[55] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[56] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in Neural Information Processing Systems*, 2015.
[57] L. Bai, L. Yao, S. Kanhere, X. Wang, and Q. Sheng, "Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *IJCAI International Joint Conference on Artificial Intelligence*, 2019.
[58] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

Xiaoming Liu received the B.S. degree in Automation from Xi'an Jiaotong University, China in 2012; and the Ph.D. degree in Cyber Science and Engineering from Xi'an Jiaotong University, China in 2019. He was a research scholar in Georgia Institute of Technology from 2017 to 2018. He is currently an Associate Professor in the School of Cyber Science and Engineering of Xi'an Jiaotong University. His research mainly focuses on Big Graph Mining, Large-Scale Heterogeneous Data Analysis and Mining, Machine Learning and its Applications.

Zhanwei Zhang is currently pursuing his master degree in the School of Cyber Science and Engineering, Xi'an Jiaotong University, China. His current research interests include machine learning, big graph mining, network theory and its applications.

Lingjuan Lyu is currently a senior research scientist and team leader in Sony AI. She was an expert researcher in Ant Group, Research Fellow with the National University of Singapore, and Research Fellow (Level B3, same level as lecturer/assistant professor) with the Australian National University. She received a Ph.D. degree from the University of Melbourne in 2018. She was a winner of the IBM Fellowship program (50 winners Worldwide) and contributed to various professional activities. Her current research interests span distributed machine/deep learning, privacy, robustness, fairness, and edge intelligence. She has publications in NeurIPS, ICLR, IJCAI, SIGIR, EMNLP, TII, JSAC, JIOT, TPDS, TDSC, etc. Her paper won the best paper award in FL-IJCAI'20.

Zhaohan Zhang is currently pursuing his master degree in the School of Cyber Science and Engineering, Xi'an Jiaotong University, China. His current research interests include Online Social Network Adversarial Technology and Machine-Generated Text Detection.

Shuai Xiao is currently a research scientist at Alibaba Group. He obtained the Ph.D. degree of School of Electronic Information and Electrical Engineering from Shanghai Jiao Tong University. Before that, he received the B.E. degree in Electronic Engineering from Huazhong University of Science and Technology in 2013. He has been awarded a scholarship by China Scholarship Council to take Joint PhD program at Georgia Institute of Technology. His research interests include machine learning, data mining, and behavior analysis.

Chao Shen (S'09-M'14) received the B.S. degree in Automation from Xi'an Jiaotong University, China in 2007; and the PhD degree in Control Theory and Control Engineering from Xi'an Jiaotong University, China in 2014. He was a research scholar in Carnegie Mellon University from 2011 to 2013. He is currently a Professor in the Faculty of Electronic and Information Engineering, Xi'an Jiaotong University of China. He serves as the Associate Dean of School of Cyber Science and Engineering of Xi'an Jiaotong University. He has published more than 50 research papers in international referred journals and conferences.

Philip S. Yu received the B.S. Degree in E.E. from National Taiwan University, the M.S. and Ph.D. degrees in E.E. from Stanford University, and the M.B.A. degree from New York University. He is a Distinguished Professor in Computer Science at the University of Illinois at Chicago and also holds the Wexler Chair in Information-Technology. Before joining UIC, Dr. Yu was with IBM, where he was manager of the Software Tools and Techniques department at the Watson Research Center. His research interest is on big data, including data mining, data stream, database and privacy. He has published more than 1,200 papers in refereed journals and conferences. He holds or has applied for more than 300 US patents. Dr. Yu is a Fellow of the ACM and the IEEE. Dr. Yu is the recipient of ACM SIGKDD 2016 Innovation Award for his influential research and scientific contributions on mining, fusion and anonymization of big data, the IEEE Computer Society's 2013 Technical Achievement Award for "pioneering and fundamentally innovative contributions to the scalable indexing, querying, searching, mining and anonymization of big data", and the Research Contributions Award from IEEE Intl. Conference on Data Mining (ICDM) in 2003 for his pioneering contributions to the field of data mining. He also received the ICDM 2013 10-year Highest-Impact Paper Award, and the EDBT Test of Time Award (2014). He was the Editor-in-Chiefs of ACM Transactions on Knowledge Discovery from Data (2011-2017) and IEEE Transactions on Knowledge and Data Engineering (2001-2004).