ST-MFM: A Spatiotemporal Multi-modal Fusion Model for Urban Anomalies Prediction

Ruiqiang Liu 1 and Shuai Zhao 1,* and Bo Cheng 1 and Hao Yang 2 and Haina Tang 3 and Fangfang Yang 1

Abstract. Urban anomaly prediction is of great importance for urban management and public safety. Accurate anomaly prediction can avoid much unnecessary loss. Urban anomalies are usually caused by many complex factors, such as festivals, demonstrations and market promotions. It is not possible to predict anomalies from the perspective of reason, thus, most of the previous work analyzes the impacts of anomalies from multiple crowd flow datasets and observes the shift to ordinary distribution when they occur. Most existing models use observation-based methods to extract relevant spatiotemporal features, which are difficult to fully extract hidden relationships and eventually lead to low accuracy and low recall. In this paper, we propose an end-to-end deep learning based approach, called spatiotemporal multi-modal fusion model to collect the impacts of urban anomalies on multiple crowd flow datasets and predict anomalies in each region of the city for next time interval in turn. More specifically, we model the city into a graph and regard each region as a node. We use graph convolution network to obtain its spatial features and use gate recurrent units to obtain its temporal features. The features of those multiple modalities are further aggregated with points of interest in a two-stage-fusion method for assigning different weights to different functional regions. We evaluate our method using five datasets associated with New York City: 311 complaints, taxicab data, bike rental data, points of interest and road network dataset. Results show the advantages nearly 10% beyond the state-of-the-art urban anomalies prediction methods.

1 Introduction

Urban anomalies including noise, illegal park, illegal assembly or illegal use of public facilities, etc. have potential threats and sometimes may pose tremendous risks to public property or safety if they are not handled timely or properly. An accurate anomaly prediction can make the city managers warned in the early stages of anomalies and people would have more time to prepare for the anomalies. In this paper, we study the urban anomaly prediction problem with multiple crowd flow data; that problem being how to predict anomalies in all the regions of the city in a future timestamp by using multiple crowd flow datasets and the points of interest (POI) statistics of each region in the city. It is unrealistic to analyze these impacts from the perspective of cause as anomalies can be caused by many factors. So we analyze the impacts of multiple datasets from anomalous events and observe the shift to ordinary distribution when they occur to predict the urban anomalies. Most existing methods are based on observation to extract relevant features [18, 4], which are difficult to fully extract hidden relationships and eventually lead to low accuracy and low recall. However, several technical challenges exist in solving the anomaly prediction problem when using deep learning based methods.

• Data sparsity: Urban anomaly records are mainly reported by the public, but people may not report anomalies or events all the time at all places in the city when they happened. So the records obtained are only a subset of the real anomalies in the city. It would deteriorate the accuracy of the model’s final prediction. Besides, crowd flow data are often severely sparse. These insufficient datasets will have great harm to the training as it is hard to get the ordinary distribution from insufficient dataset especially when using deep learning based method.

• Feature-based fusion: Most of the urban computing problems are based on the historical observation of a dataset to predict the next state of the same dataset or simply predict a binary classification. In this paper, we analyze the impacts of urban anomalies by deviating other datasets from ordinary distribution, so we need to fuse the features extracted from multiple datasets. This can easily lead to over-fitting problems. How to find a proper way to combine features from datasets in different sources and distribution is a challenge.

• Complex spatial relationship: Urban regions are not regular grids, and the outlines of the cities are also not regular. Typical methods usually model the city into a grid map and ignore the road network relationships and the outline of regions, only by considering the relationship between the regions from the Euclidean distance [19]. In fact, the division of each region in the city is mostly divided by both the road network system and some geographical factors like hills or rivers within the city. Two regions with long Euclidean distances may also have a great impact on each other as there may be a highway connection. How to combine the road network system to consider the connection between regions is also a challenge.

To address the issues mentioned above, we propose a deep learning method based multi-modal framework named spatiotemporal multi-modal fusion model (ST-MFM for short) to predict anomalies in all the regions of the city. In order to consider the road network relationships and POI characteristics, we first model the city as a graph and consider each region as a node and POI in regions as the features of nodes. Then we obtain the spatial and temporal information of each crowd flow dataset by using graph convolution network (GCN).
and gate recurrent units (GRU). Finally we design a two-stage fusion method to fuse all the information obtained from multiple datasets to predict the region where the next anomaly will occur. We use 5 real-world datasets collected from the city of New York, including bike rental dataset, taxi cab dataset, POI dataset, road network system dataset and 311 complaints dataset to evaluate our method.

The evaluation results show that our model can predict anomalies in all regions timely and more accurately than the state-of-the-art baselines. In summary, our contributions are summarized as follows:

- We designed a unified spatiotemporal multi-modal fusion model that jointly considers multiple crowd flow datasets, road network system dataset and POI dataset to predict the urban anomalies. Analysis from multiple aspects will be more analytical than using only single modal data, and it can compensate for the impact of data sparseness to some extent.
- We proposed a modeling approach to model the city as a graph, and a GCN-based model that captures the spatial features of crowd flow dataset. We found that approach is a better way than rasterizing the city into a grid map and using CNN to obtain spatial features when handling the urban ground plane problems.
- We conducted two parts of the extensive experiments on real-world datasets from the city of New York. The results confirm that our modeling approach is better than rasterizing, and also our method consistently outperforms the competing baselines by more than 10%.

2 Related Work

2.1 Spatiotemporal prediction in urban computing

Many methods have been designed for predicting urban anomalies previously. In [22, 18], both the authors divide urban anomalies predicting into two states, which can be summarized as probing and aggregation. And also a Markov model is designed to predict the current state of each region by considering multiple previous states to predict urban events [4]. Though deep learning has proven its performance and efficiency in various fields of urban computing, none of their models is deep-learning-based. In recent work, deep learning based algorithms have also appeared to predict urban anomalies. Considering that the anomalous events are not enough to train a deep neural network, the authors turn to predict the normal crowd flow and compare it with the actual flow to predict whether an abnormal event has occurred [20].

Dataset in urban computing usually has spatial and temporal attributes, while problems based on spatiotemporal prediction are fundamental for data-driven urban management. A large number of methods have been developed to various topics in prediction problem of urban computing, including the taxi cab or ride-hailing demand [4, 16], predicting crowd flow data [17, 15, 23, 11, 5], and traffic prediction [21]. These problems are similar to some extent, which are all to extract the temporal and spatial attribute distribution rules embedded in the datasets through deep-learning-based models. However, their tasks are all based on a single modality, which uses historical data from a dataset to learn its spatiotemporal distribution and predict the future timestamp of the same dataset.

The crowd flow predicting problem is similar to the task of our submodality network, which is also extracting the spatiotemporal features in crowd flow datasets. CNN-based residual networks are used to capture spatial dependencies and three categories including closeness, period, and trend to capture its temporal dependencies [19]. Furthermore, spatiotemporal features can be extracted by local CNN and LSTM model [15] and convolutional recurrent network which is combined with CNN and RNN [23]. In all the methods mentioned above, a city is first rasterized into a grid map, and CNN is used to capture spatial dependency which we thought is not so properly. We would explain the reason later and confirm it with some experiments.

2.2 Graph convolution network in urban computing

Graph convolution network can be seen as an operator operating on the graph [7]. It allows data from non-Euclidean structures to retain graph topology information and share parameters like CNN. GCN is ideal for dealing with urban computing problems on the ground plane, as the road system network divided a city into a graph with regions naturally. Recent researches in urban computing have used GCN to extract spatial dependencies. Multi-graph convolution network work is used to predict ride-hailing demand by constructing multiple graph relationship between regions and extract the features with GCN [3]. However, they also divide the city into a grid map and consider a grid as a region, instead of dividing the city with road network system. In traffic prediction problem, the linear operation in GRU is replaced with graph convolution operation and each road is considered as a vertex to extract the spatiotemporal relationships [21]. Both their tasks and modeling ideas are different from ours, and they use historical data from the same dataset to predict the future timeslot condition.

2.3 Deep learning based multimodal fusion framework

Problems in urban computing are often affected by multiple factors, and methods based on multi-modal fusion usually get a better performance compared with the models that analyse from a single dimension of datasets. The urban big data fusion methodologies can be summarized by classifying them into three categories: DL-output-based fusion, DL-input-based fusion and DL-double-stage-based fusion [9]. DMVST-Net is proposed to extract the features from both spatiotemporal view and semantic view and fuse the outputs of those submodals [15]. In [12], the temporal feature and spatial feature are first extracted from the same dataset and used as input to fuse into the model to profile urban residents. All their jobs are to use the same dataset to extract different dimensions and merge to predict the state of the same dataset in the next moment or binary classification problem, which is fundamentally different from the way we use multiple datasets to analyze the shift from ordinary distribution to predict the next state of another dataset. Those difficulties make the urban anomaly prediction problem previously solved by analyzing the state rather than an end-to-end deep learning network. Though the deep learning network has proved its performance in many other problems.

3 Preliminaries

In this section, we formulate the problem of urban anomaly prediction using multiple crowd flow datasets and other external datasets, including POI and road system datasets.
3.1 Definition: Urban ground plane graph

In this study, we use road network system dataset including highways and arterial roads and also geography information of the city to partition a city into regions with a map segmentation method. We then consider each region \( r_i \mid r_i \in \{r_1, r_2, ..., r_v\} \) as a node in graph, \( v \) means the number of the regions in the city, as the graph has \( v \) nodes. If two regions are adjacent to the boundaries or reachable directly by road system like bridges and tunnels without passing through another region, these two nodes in graph are considered to be connected. The adjacency matrix of the graph can be define as \( A \in \mathbb{R}^{v \times v} \) where

\[
A_{i,j} = \begin{cases} 
1, & r_i \text{ and } r_j \text{ are connected} \\
0, & r_i \text{ and } r_j \text{ are not connected}
\end{cases}
\]

From the POI dataset, we can get the number of hospitals, parks, etc. in each region of the city. It can be denoted as a tensor \( F \in \mathbb{R}^{v \times f} \), where \( f \) means the number of POI categories. We recognize that its POI characteristics reflect the functional characteristics of the region, and we consider that crowd flows in regions with similar functional characteristics will have a close flow pattern [13]. For example, the work-oriented area will have a similar peak period of time, i.e., if two regions are adjacent to the boundaries or reachable directly by road system like bridges and tunnels without passing through another region, these two nodes in graph are considered to be connected.

3.2 Definition: multiple crowd flow datasets (inflow/outflow)

In this study, we use bike rental dataset and taxi pickup-pickoff dataset as multiple crowd flow datasets. A crowd flow dataset \( \Gamma \) consists of a bunch of trip records, each of which can be considered as a six-element tuple \( < t_s, lat_s, lng_s, t_e, lat_e, lng_e > \), meaning the timeslot and location of beginning and ending point of the trip. The subscripts of \( s \) and \( e \) mean the beginning point and ending point of the trip, and \( lat, lng \) mean the latitude and longitude of the points.

At the timeslot \( t \), inflow and outflow in all nodes can be denoted as a tensor \( x_t \in \mathbb{R}^{2 \times n} \), where \( x_t^{in} = X_t^{in}, x_t^{out} = X_t^{out} \).

3.3 Definition: anomalies prediction problem

Similarly, we count the dataset of urban abnormal reports in all regions of the city. A threshold is set in advance, and when the number of abnormal reports in a region exceeds the average number of anomalies in the region by more than this threshold within a certain interval of time, it is considered that an anomaly occurs in this region at this time interval. The anomalies at the \( t^{th} \) interval of all regions as the prediction target can be denoted as \( Y_t \in \mathbb{R}^{1 \times v} \). So the anomalies prediction problem can be redefined as:

Given the historical observations in \( N \) intervals of \( n \) crowd flow datasets \( \{X_{1,t}, X_{2,t}, ..., X_{m,t} \mid t = T, T - 1, ..., T - N \} \), the features of all nodes \( F \), and the adjacency matrix of the graph \( A \), to predict \( Y_T \).
4 Proposed Spatiotemporal Multimodal Fusion Framework

In this section, we provide details for our spatiotemporal multimodal fusion model. Figure 3 shows the architecture of our proposed method, which is comprised of two stages of processing. The first status of processing is to extract spatiotemporal dependencies from each crowd flow dataset. It contains three subnetworks, each of which is motivated by the combination of GCN and GRU neural networks. In the second stages of processing, we fuse those modality features with POI features, and use a shared-weight fully connected layer to get the final prediction result.

4.1 Spatiotemporal subnetwork

As we mentioned above, most of the work in the extraction of spatial dependency uses the method of dividing the city into a grid map and extract it with CNN. Here we propose a new modeling method that uses road network system to separate the city into a map of graph, and use GCN to extract spatial information. Below we quantitatively analyze why our method is superior to CNN-based method in urban ground plane computing problem.

Figure 4 shows how CNN and GCN work. The convolution of CNN is essentially a filter that uses shared parameters. The feature map is implemented to calculate the spatial feature by calculating the weighted sum of the central pixel and the adjacent pixel. However, non-transform invariance cannot be maintained on the non-Euclidean structure data, as the topology of a graph cannot guarantee the same number of adjacent nodes from each node. This mechanism makes CNN unable to process non-Euclidean structure, e.g. graph. And if CNN is used to deal with the urban problem of the ground plane, it means to ignore both the relationships between regions and also outlines of regions but forcibly divide the city into rectangular areas. Those methods including CNN and local CNN would destroy the topological relationships between regions of the city. Furthermore, the CNN operator will put the nearby non-information space, such as the space within river, into the calculation and discard the relevant regions such as the regions at the other end of the bridges.

The convolution in GCN is more like an operation on the graph [8]. With the help of eigenvalues and eigenvectors of Laplacian matrices, GCN can extract the spatial dependency of the topology graph instead of destroying it. With normalized Laplacian matrix \( L = I - \frac{1}{\sqrt{2}} A D^{-\frac{1}{2}} \), \( A \in \mathbb{R}^{n \times n} \) is the adjacency matrix, and \( D \) is the degree matrix of adjacency matrix \( A \). Graph convolution operation [2] can be defined as

\[
X_{l+1} = \sigma \left( \sum_{k=1}^{K} \alpha_k L^k X_l \right) \tag{4}
\]

where \( X_l \) denotes the features in \( l \)-th layer, \( \alpha_k \) is the trainable variable, and \( \sigma \) is the activation function. In terms of the convolution operation, \( K \) actually defines the size of the receptive field. If \( K \) is set to be 1, the filters from GCN act on each node of the graph and its first-order neighbourhood. If \( K \) is set to be 2, they would act on both its first-order and second-order neighbourhood and calculate the weighted sum. So different levels of travel in multiple crowd flow dataset should set different values of \( K \) as they have obviously different probability of crossing the number of regions. For example, \( K \) of the taxi dataset should be smaller than the \( K \) of the bike dataset as taxi passengers are more likely to reach the final destination directly instead of continuing to move to another area. Therefore taxi passengers will have a smaller scope of influence in other surrounding areas than the bicycle traveler at the same time and same location.

Extracting both the temporal dependence and spatial dependence is a key problem in handling crowd flow data. Considering that the crowd flow data is not only closely related to the data at the previous time intervals, it also has periodicity at the same time of both the previous days and weeks. We first pick out the relevant consequent historical observation, including \( \{X_{t-1}, ..., X_{t-n_1}\} \), \( \{X_{t-24}, ..., X_{t-24-n_2}\} \), \( \{X_{t-(24 \times 7)-1}, ..., X_{t-(24 \times 7)-n_3}\} \) representing the situation a week ago, a day before and at this moment. \( n_1, n_2 \) and \( n_3 \) are three hyperparameters, which are length of closeness, period and trend sub-model inputs. These three inputs would be sent into three sub-models as the categories of closeness, period and trend with the same network structure, including a k-hop GCN and several GRU units, to extract a longer temporal dependency of the distribution. The k-hop GCN will extract the spatial distribution features of the crowd flow data in each region from itself to the k-order neighbor regions.

\[
X_{t+1}^k = \sigma(f(L^k) X_t^k W^k) \tag{5}
\]

\( f(L^k) \) is the k-order polynomial function of the Laplacian \( L \), to approximate the connection relationship of regions in k-hop. \( X_t^k \) means the input in \( t \) interval of \( l \) layer, and \( W^k \) is the learnable parameters. Then the vector that fully extracts the spatial features will be sent to GRUs to capture the temporal features of successive time intervals.

We use GRU network to capture the temporal sequential dependency, which is proposed to address the exploding and vanishing gradient issue of traditional Recurrent Neural Network (RNN). In this paper, we use the original version of GRU [1] and formulate it as:

\[
h_t = GRU(X_{t+1}^k; h_{t-1}) \tag{6}
\]

where \( h_t \) is the output representation of all regions at time interval \( t \).

4.2 Multi-modal fusion

We consider that the tensor merged from early fusion in each modality has fully extracted the temporal and spatial variation of each region itself and its adjacent regions through spatiotemporal subnet. However, our ultimate task is to predict urban anomalies. An anomaly might not be that anomalous in terms of a single dataset but might be considered as an anomaly when checking multiple datasets simultaneously as we are predicting by accumulating the deviation of multiple datasets. Besides, due to the difference in regional functionality, a pattern of crowd flow change that is abnormal in one region may not be an abnormality but a normal behavior in another region. For example, a large number of crowds entering a work-oriented area on Friday afternoon may be because there is a large event happening. But it may be just a normal situation if this pattern occurs in an entertainment-oriented area. So we need to combine these extracted spatiotemporal features from multiple crowd flow data with the functionality of specific regions, to finally predict the probability of urban anomaly in all regions.

In late fusion component, we fuse the tensor of both features extracted from multiple modalities and POI and use fully connected layers to get the final prediction. We first add these features to each node directly in the graph with the method of adding channels, as these features are all information on some kinds of dimensions of the region. Considering that the features of each graph node already contain the spatiotemporal features of its own and neighbour nodes...
and also the regional function information, for this fully connected layer, it needs to do the same job for each node. That is, use all these information about this region to analyse whether an abnormality will occur no matter which corner of the city the region is in the city. So we design the parameters in this fully connected layer to be shared by all the nodes in the graph and choose the sigmoid function as the activation function to make the output in the range of [0, 1] to indicate the probability of anomalies in this region.

\[ Y_t = \text{sigmoid}([h_i | i \in m]; F] \times W_{\text{shared}} + b_{\text{shared}}) \]  

(7)

\[ W_{\text{shared}} \in \mathbb{R}^{n \times 1} \] and \( b_{\text{shared}} \in \mathbb{R}^{1 \times 1} \) is learnable parameters, \( a \) is the sum of channels of all spatiotemporal features from \( m \) modalities and channels of features from POI \( F \). \( \{h_i | i \in m\} \) means the collection of \( m \) modality features and \([;] \) means the concatenation of the collection.

In our work, we predict the probability of urban anomaly in every region in the city in \( t \) interval, the loss function is defined as:

\[ L(\theta) = \|Y_t - \hat{Y}_t\| + \lambda L_{\text{reg}} \]  

(8)

The second term \( L_{\text{reg}} \) is an L2 regularization term to minimize the size of the network model and avoid the over-fitting problem, and \( \lambda \) is a hyperparameter. \( \theta \) are all learnable parameters in ST-MFM.

### 4.3 Algorithm and Optimization

Algorithm 1 outlines ST-MFM training process. We first construct the training instances from the original sequence data, ST-MFM is trained via backpropagation and Adam optimizer [6].

**Algorithm 1 ST-MFM Training Algorithm**

**Input:** Historical observation of \( m \) crowd flow datasets:

\( \{X_{1,t}, X_{2,t}, ..., X_{m,t} | t = T, T-1, ..., T-N\} \); 

historical records of anomalies: \( \{Y_{T-N}, ..., Y_{T-1}\} \); 

features of all nodes: \( F \); 

adjacency matrix of the graph: \( A \); 

lengths of closeness, period, trend sequence: \( n_1, n_2, n_3 \).

**Output:** Learned ST-MFM model

1: \( D \leftarrow \emptyset \)
2: for all crowd flow datasets \( i (i \leq i \leq m) \) do 
3: for all available time interval \( t (1 \leq t \leq N - 1) \) do 
4: \( S_c = \{X_{i,t-1}, ..., X_{i,n_1}\} \)
5: \( S_p = \{X_{i}-(24)_{t-1}, ..., X_{i}-(24)_{n_2}\} \)
6: \( S_o = \{X_{i}-(24)_{t-1}, ..., X_{i}-(24)_{n_3}\} \)
7: put an training instance \((S_c, S_p, S_o, Y_t)\) into \( D \)
8: initialize all learnable parameters \( \theta \) in ST-MFM
9: repeat 
10: randomly select a batch of instances \( D_b \) in \( D \)
11: find \( \theta \) by minimizing the objective (6) with \( D_b \)
12: until stopping criteria is met

### 5 Experiments

In this section, we introduce two parts of our experiments with five real-world datasets collected in New York City. The first part of the experiment is to verify that we model the city into a graph and used GCN to extract spatial dependency is better than those methods
which model the city into a grid map and use CNN to extract the spatial dependency. Therefore, we use a spatiotemporal subnetwork in the model, i.e., the NYC bicycle rental dataset or the NYC taxi dataset, to predict the crowd flow at the next time interval, as there is more deep-learning-related work on this issue. In the second part of the experiment, we evaluate our urban anomaly prediction performance of our model with several existing urban anomaly prediction methods.

5.1 Experiment Settings

Datasets We evaluate our proposed model with five real-world datasets from New York City. Each dataset details are as follow.

- **NYC-Bike**: NYC-Bike dataset contains 8081216 trip records with 344 bike stations and more than 6000 bikes of NYC in 2014, from 01/01/2014 to 01/01/2015.
- **NYC-Taxi**: NYC-Taxi dataset contains 165 million trip records and more than 14 thousands taxicab of NYC from 01/01/2014 to 01/01/2015. The dataset including both yellow taxi and green taxi in NYC.
- **Road Network**: We collect the road network dataset in NYC including level from $L_1$ to $L_5$ in NYC, 2013. To simplify the problem, we approximate road information did not change in 2014.
- **NYC-Anomaly**: We use 311 data in NYC as the anomalies report dataset. 311 is NYC’s governmental non-emergency service number, which allows people in the city to complain about everything. The dataset is a sub-dataset with four kinds of 311 Service Request including "Noise", "Blocked Driveway", "Building/Use", "Illegal Parking" in 2014. And we set the threshold for the anomalous reports to be 5. That is, when the number of anomalous reports in a region exceeds the average number of anomaly reports in the region by more than 5, we consider an anomaly occurs in this region at this time interval.

5.2 Results on crowd flow prediction

We take our spatiotemporal subnetwork as a GCN-GRU based method to predict the crowd flow to evaluate the capacity for extracting spatiotemporal relationships because this sub-topic has more outstanding work to be compared with. We only use the NYC bicycle rental dataset or the NYC taxi dataset alone as a submodal input, and add a fully connection layer with sigmoid activation function in the output of submodel to make its output range $[-1,1]$. Finally, the output is denormalized as the final prediction of regional changes in crowd flow.

In this part of experiment, we measure our method by Root Mean Squared Error (RMSE) as

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\hat{x}_i - x_i)^2} \quad (9)$$

where $\hat{x}_i$ is the denormalized predicted value and $x_i$ is the actual flow change value as $\Delta_{inflow} - \Delta_{outflow}$ in region $i$. We compare our method with the following methods:

- Auto-Regressive Integrated Moving Average (ARIMA): ARIMA is a well-known model for understanding and predicting future values in a time series.
- Vector Auto-Regressive (VAR): VAR is a more advanced spatiotemporal model that captures the pairwise relationship between all flows. But due to the large number of parameters, it has heavy computational costs.
- Convolutional LSTM (ConvLSTM) [10]: ConvLSTM is a deep learning model that combines CNN and LSTM, specifically designed for spatiotemporal prediction. It replaces the convolution operation with the switch between states.
- Deep Spatio-Temporal Residual Networks (ST-ResNet) [19]: ST-ResNet extracts the spatiotemporal features by using residual learning and merges them in a fusion process along with external information such as weather condition and holidays.
- Deep Multi-View Spatial-Temporal Network (DMVST-Net) [16]: DMVST-Net is a multi-view based deep learning approach. It consists of three different views: the temporal view, the spatial view, and the semantic view modeled with LSTM, CNN and graph embedding respectively.

Table 2 shows the performance of our proposed method comparing with all other competing methods. We run each baseline 10 times and report the mean and standard deviation of each baseline.

Table 1 shows the details of actual flows as $\Delta_{inflow} - \Delta_{outflow}$ from two datasets, including the maximum value, minimum value and variance of all time intervals in all regions. These data can be used to measure the accuracy of the crowd flow prediction. We choose data from the last two months as the testing data, and all data before that as training data.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total</th>
<th>Maximum</th>
<th>Minimum</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>NYC-Bike</td>
<td>8M</td>
<td>186.30</td>
<td>-174.00</td>
<td>8.67</td>
</tr>
<tr>
<td>NYC-Taxi</td>
<td>16SM</td>
<td>1703.83</td>
<td>-2740.00</td>
<td>1533.67</td>
</tr>
</tbody>
</table>

In these methods except GCN-LSTM we all use the modeling approach that rasterizes the city into rectangular areas with a grid of multiple sizes including $20 \times 20$, $(40,40), (29,29)$ to reduce the impacts of grid sizes. We choose the grid of these sizes in order to make the total number of regions roughly the same as our modeling method. It ensures that the actual results of crowd flow are distributed similarly in different modeling styles.

Table 2 shows the performance of our proposed method comparing with all other competing methods. We run each baseline 10 times and report the mean and standard deviation of each baseline. Specifically, the traditional time-series prediction methods do not perform well, as these methods only focus on the law at the same time intervals of history or simply calculate the mean. Other external factors and inter-regional spatial connection have been ignored. For deep learning algorithms that take into account both the regions spatial relationships and the temporal dimension from regions, these methods are better than those traditional time series prediction method.

Also, our method outperforms those neural-network-based methods. Compared to those models, our network is not too deep and size...
is not too large. We consider this is because the crowd flow dataset we use to predict is traveled through the road network system. Dividing the city by road network system may be more relevant to the real situation, so this modeling method can obtain more accurate spatial features. And GCN method can extract only the information of the most relevant closely related neighbor regions and ignore the distant regions that are not closely related. The results confirm that our modeling approach is more suitable for solving urban ground plane computing problem as it can extract the spatiotemporal features better.

5.3 Results on urban anomalies prediction

Considering that there are many factors involved in the urban forecasting problem, and it is also not enough to predict the occurrence of urban anomalies by analyzing the effects on a single dataset bring from anomalies. Most of the existing methods need to manually extract features of divide the calculation into multiple parts rather than an end-to-end deep learning based network. We compare our ST-MFM with the following baselines:

- **Crowdsourcing-based Urban Anomaly Prediction Scheme (CUAPS) [4]**: CUAPS develops a Bayesian model to identify anomalies distribution and uses a Markov model to predict anomalies.
- **Urban Anomaly PreDiction (UAPD) [14]**: UAPD focuses on the distribution of anomalous reports and detects the change point of the anomaly sequences to predict anomalies.
- **Compute and Aggregate individual anomaly scores (ind+int) [18]**: It proposes the similarity-based “Compute Individual Anomaly Scores” to give an anomaly score and detects anomalies by OC-SVM and rbf kernel.

### Table 2. Results on crowd flow prediction

<table>
<thead>
<tr>
<th>methods-(grid size)</th>
<th>Bike-RMSE</th>
<th>Taxi-RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA-(40,20)</td>
<td>8.14</td>
<td>25.51</td>
</tr>
<tr>
<td>VAR-(40,20)</td>
<td>8.20±0.33</td>
<td>33.53±6.37</td>
</tr>
<tr>
<td>ConvLSTM-(29,29)</td>
<td>5.31±0.63</td>
<td>16.13±5.74</td>
</tr>
<tr>
<td>ST-ResNet-(40,20)</td>
<td>4.52±0.42</td>
<td>15.74±2.54</td>
</tr>
<tr>
<td>DMVST-Net-(40,20)</td>
<td>3.88±0.48</td>
<td>12.35±3.32</td>
</tr>
<tr>
<td>GCN-GRU</td>
<td>2.40±0.56</td>
<td>10.15±2.14</td>
</tr>
</tbody>
</table>

Table 2 shows the result of our method and other baseline methods. From the experimental result, our method is better than other methods. Our approach is an end-to-end process where the associated features between data and data are discovered by the model training rather than relying on manual observation. This makes some hidden relationships that are difficult to find through observation but also useful can also be extracted as features, and the robustness of the algorithm is better than those of non-deep-learning-based methods. The method which focuses on the distribution of anomaly dataset does not perform well. We think that may be because the anomaly distribution of each city or region is different. Focusing on the distribution of the anomaly itself may lead to reduced generalization of the model.

5.4 Effects of different components

To investigate the effects of different components, we evaluate the following variants of ST-MFM by removing or changing different components from the model, including:

- **ST-MFM-NoPeriod (NoPeriod)**: We transform the spatiotemporal features extraction module by removing long temporal dependencies and leave only the closeness module and expand its input to the original input number. That is to set \( \{ n_1, n_2, n_3 \} = \{ 0, 0, 12 \} \).
- **ST-MFM-NoPOI (NoPOI)**: NoPOI removes the late fusion stage, which uses POI data to fuse in order to consider different anomalies performance in different functional regions. It inputs the results of the early fusion phase into the activation function of \( \text{tanh} \) and directly obtains the anomalies prediction results.
- **ST-MFM-CNN (ST-CNN)**: ST-CNN models the city into a grid map and replaces the GCN structure by CNN.
- **ST-Bike**: ST-Bike model removes the multiple modality fusion stage and only uses the spatiotemporal extraction module with bike rental dataset and simply fuse POI to predict the anomalies.
- **ST-Taxi**: Similar to ST-Bike, ST-Taxi only use the taxicab dataset and POI to predict the anomalies.

### Table 4. Results on urban anomalies from NYC 311 Services with different components

<table>
<thead>
<tr>
<th>methods</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-MFM</td>
<td>0.740</td>
<td>0.725</td>
<td>0.803</td>
<td>0.788</td>
</tr>
<tr>
<td>NoPeriod</td>
<td>0.722</td>
<td>0.703</td>
<td>0.805</td>
<td>0.780</td>
</tr>
<tr>
<td>NoPOI</td>
<td>0.704</td>
<td>0.663</td>
<td>0.760</td>
<td>0.722</td>
</tr>
<tr>
<td>ST-CNN</td>
<td>0.660</td>
<td>0.688</td>
<td>0.720</td>
<td>0.705</td>
</tr>
<tr>
<td>ST-Bike</td>
<td>0.700</td>
<td>0.691</td>
<td>0.766</td>
<td>0.677</td>
</tr>
<tr>
<td>ST-Taxi</td>
<td>0.655</td>
<td>0.644</td>
<td>0.683</td>
<td>0.632</td>
</tr>
</tbody>
</table>

Table 4 presents the results of the proposed measures. i) The performance of ST-CNN is much worse than other multi-modal fusion models, which implies again that the graph modeling approach contains more useful information than rasterizing the city into rectangular regions. ii) The performance of NoPeriod is close to but still worse than ST-MFM, which indicates that the temporal information is concentrated in the most recent time period, and also indicates the efficiency of longer temporal dependency module. iii) The performance of ST-Bike and ST-Taxi is worse than NoPeriod and NoPOI, which certifies the significance of multi-modals. iv) ST-MFM achieves the best performance, which indicates the graph modeling approach as well as multi-modal component could effectively extract features and fuse them to improve the prediction performance.

6 Conclusion and Discussion

In this paper, we investigate the urban anomalies prediction problem and we are the first to define the city into a graph by using road network system and propose a deep learning based end-to-end method to solve this problem. We propose a spatiotemporal multi-modality fusion model which uses graph convolution network and gate recurrent units to extract the spatial features and temporal features from multiple crowd flow datasets, and finally fuse them with POI to predict
the probabilities of anomalies. We further demonstrate why modeling the city into a graph and using graph convolution network is more suitable than CNN in urban ground plane problem and we do comparative experiments to confirm the idea. When evaluated on real world datasets, the proposed model achieved significantly better results than other existing state-of-the-art methods. For future work, we plan to (1) consider the levels of the road system and assign different weights in the graph with different levels of roads when modeling; (2) aggregate other graphic processing algorithms in the model; (3) evaluate the proposed model on other urban computing forecasting tasks.

7 Acknowledgments

This work is supported by Beijing Natural Science Foundation (Grant No. 4182042), Beijing Nova Program of Science and Technology (Grant No. Z1911000119031), National Key Research and Development Program of China (Grant No. 2018YFB1003804), and The Open Program of Zhejiang Lab (Grant No. 2019KE0AB03).

REFERENCES


