2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress

ST-TAP: A Traffic Accident Prediction Framework Based on Spatio-Temporal Transformer

Weitao Liu, Xuanyi Liu, Hui Feng, Yiran Wang
Zhejiang Energy-R&D
Hangzhou 311121, China
{liuweitao,liuxuanyi}@zjenergy.com.cn,
{532221103, 417465065}@qq.com,

Lintao Guan, Weifeng Xu, Guojiang Shen, Zhi Liu, Xiangjie Kong
College of Computer Science and Technology,
Zhejiang University of Technology,
Hangzhou 310023, China
ltguan1996@outlook.com, 2207394788@qq.com,
{gjshen1975, lzhi}@zjut.edu.cn, xjkong@ieee.org

***Abstract*—As an important part of urban management, researchers have made many efforts in accident prediction. The current research mainly considers the impact of the temporal features of traffic flow and the fixed topology of the road on the accident. However, these studies do not consider the changes in the relationship between road spatial features and accidents over time. In order to better achieve the accident prediction, we first take the traffic feature parameters of traditional highways traffic fixed station data after pre-processing as input dataset. Then, we use ClusterGAN to combine traffic spatio-temporal features to cluster the dataset and divide it into multiple datasets. After that, we combine the temporal and spatial features of the traffic flow with the temporal and spatial Transformer to capture the static and dynamic temporal and spatial dependence of the traffic flow, and obtain the accident probability through multiple extractions of temporal and spatial dependence of the traffic flow and convolutional neural network to accurately predict traffic accidents in real time. Finally, we compared with a variety of existing methods to verify the advance of this method.**

Keywords—intelligent transportation, deep learning, Transformer, accident prediction, ClusterGAN

I. INTRODUCTION

With the development of social economy, the urban population has grown rapidly, and the number of motor vehicles has increased. What followed was the frequent occurrence of various traffic problems, which seriously affected the healthy development of the city[1]. Accurate prediction of traffic accidents[2] can increase drivers' alertness, help them focus on traffic conditions, and provide decision-making basis for traffic management departments to help them optimize road traffic control and reduce the risk of traffic accidents. In particular, with the development of cyber technology[3], the continuous improvement of the Internet of Vehicles[4] allows us to better track the trajectory of the vehicle to make the realization of accident prediction easier.

In the traditional field of accident prediction, researchers can predict the occurrence of future accidents by analyzing the existing road structure and traffic flow features[5], thereby taking advance measures to reduce traffic accidents. Essentially, the accident prediction model is a mathematical relationship used to express the relationship between the average accident frequency of a site and the traffic flow and other site features. However, the establishment of mathematical models for accident prediction requires expert knowledge[6], which has caused a large number of researchers to be unable to establish better mathematical models.

With the continuous deepening of edge computing research[7] and the continuous development of the Internet of Vehicles[8], the problem of deep learning computing resource consumption and data problems have been solved[9],[10], and its application in the field of transportation continues to deepen[11]. Compared with traditional learning structures, deep learning has the ability to use distributed and hierarchical feature representation to model complex nonlinear phenomena, and use data-driven instead of expert knowledge training models[12]. As a deep learning method, Transformer can capture the relevance in the data sequence[13]. On this basis, the improved spatio-temporal Transformer[14] can capture the time dependence and space dependence in the traffic flow sequence.

In order to solve the above challenges，we proposes a traffic accident prediction framework ST-TAP. The contributions of our work can be summarized as follows:

1. We use ClusterGAN to cluster the data before traffic accident prediction, divided the dataset and trained the model separately to improve the pertinence of the model.
2. We propose a traffic accident prediction framework based on the spatio-temporal Transformer, which improves the accuracy of the accident prediction model and reduces the model training time.
3. We use a real-world traffic accident dataset to conduct experiments to evaluate our model. Experimental results show that ST-TAP can predict accidents accurately and give corresponding risk warnings.

II. FRAMEWORK OF THE ST-TAP

In the ST-TAP, we use ClusterGAN[15] clustering on the original traffic dataset to divide the data into multiple data sets at first, and then use the accident prediction model for each dataset. The accident prediction model based on the spatial-temporal Transformer is mainly divided into four parts, which are the input layer, the spatial pre-order codec Transformer, the temporal Transformer and the prediction

978-1-6654-2174-4/21/$31.00 ©2021 IEEE
DOI 10.1109/DASC-PICom-CBDCom-CyberSciTech52372.2021.00068

layer. The overall framework of the ST-TAP is shown in Fig.**1**.

In the input layer, in order to achieve accident prediction, parallel convolutional neural networks are used in feature extraction of the time-sorted traffic speed matrix and traffic flow matrix. Use convolutional neural network to initially extract the temporal periodicity of traffic speed and flow and the spatial correlation between upstream and downstream. In order to facilitate the introduction of the following model structure, the model input is uniformly defined as a matrix **X**.

In the spatial-temporal Transformer module, the model captures the temporal and spatial features of the data. In the spatial Transformer and temporal Transformer, first learn the spatial-temporal embedding position of each node feature, initialize the spatial-temporal embedding matrix, inject the position information into the data sequence, and simulate the spatial-temporal dependence based on the adjacency matrix. Then, use the degree matrix and adjacency matrix of the node to construct a normalized Laplacian matrix in the spatial Transformer, Combining static graph convolution and dynamic graph convolution operations to capture the static spatial dependence and time-varying hidden spatial dependence in the road topology, and use the gating mechanism to fuse the spatial features of static graph convolution and dynamic graph convolution learning.

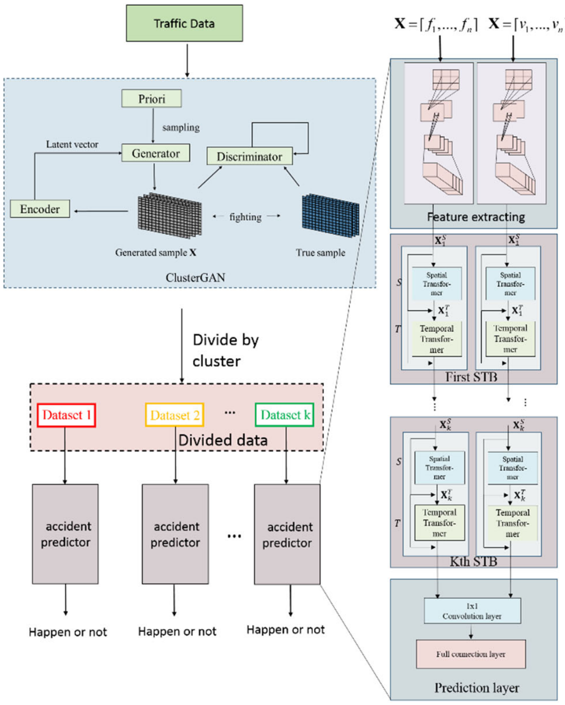


Fig.1. The framework of ST-TAP

In the temporal Transformer, the time embedding position is combined with the input parameters, and a one-dimensional vector is generated for each node at each time step and constructed into a time series, and use the self-attention mechanism combined with the sliding window to process the time series in parallel to capture the long-distance time dependence.

III. Traffic Accident Prediction Model

In the prediction of traffic accidents, how to construct the relationship between the spatial features of traffic flow and the accident is a difficult problem. In the research, it is found that the spatial features of traffic flow can be divided into static spatial-temporal dependence and dynamic spatial-temporal dependence. The dynamic spatial-temporal dependence has obvious changes in the morning and evening peaks in the congested state and the flat peak interval in the unblocked state. Among them, the static temporal and spatial dependence is mainly determined by the topological structure of the road section, the number of lanes in each section, and the connection between the road section and the ramp.

First use the spatial Transformer in the spatial-temporal Transformer block to capture the static and dynamic spatial dependence of traffic flow. Before capturing the spatial dependence, Since the input of this article is a time-labeled road upstream and downstream traffic speed matrix and traffic flow matrix, the temporal and spatial steps are different, it is necessary to ensure the simultaneous capture of the temporal and spatial features of the traffic flow at different times. In this paper, this problem is solved by implementing spatiotemporal position embedding in the spatiotemporal Transformer results. The structure of spatial Transformer and temporal Transformer are shown in Fig **2** and Fig **3**.

To embed spatial-temporal location information, firstly, the spatial location embedding matrix $\mathbf{D}^S \in \mathbb{R}^{P\times P}$ needs to be initialized according to the road topology which is used to Simulate temporal and spatial dependencies. Then the spatial location embedding matrix $\mathbf{D}^S$ is tiled along the y-axis to generate $\mathbf{D}^S \in \mathbb{R}^{P\times Q\times Q}$, and together with the tiled temporal location embedding matrix along the x-axis, the spatial-temporal location embedding feature $\overline{\mathbf{X}}^S \in \mathbb{R}^{P\times Q\times d_G}$ with fixed dimensions d_G are obtained. The calculation formula is as follows:

$$\overline{\mathbf{X}}^{ST} = F_t([\mathbf{X}^S, \mathbf{D}^S, \mathbf{D}^T]) \qquad (1)$$

where: F_t is a 1×1 convolutional layer, which is used to convert the connected features into a d-dimensional vector of each node at each time step. $\mathbf{X}^S$ is the three-dimensional tensor input of Q nodes under time step P, $\mathbf{D}^T$ temporal position embedding matrix.

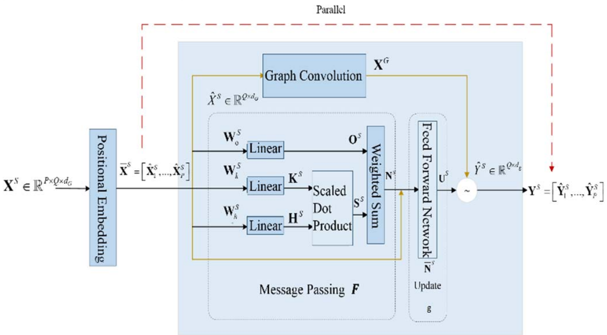

Parallel
$\mathbf{X}^S \in \mathbb{R}^{P\times Q\times d_G}$
Positional Embedding
$\overline{\mathbf{X}}^S = [\hat{\mathbf{X}}_1^S,...,\hat{\mathbf{X}}_P^S]$
Graph Convolution
$\mathbf{X}^G$
$\hat{X}^S \in \mathbb{R}^{Q\times d_G}$
$\mathbf{W}_o^S$
$\mathbf{W}_k^S$
$\mathbf{W}_h^S$
Linear
Linear
Linear
$\mathbf{O}^S$
$\mathbf{K}^S$
$\mathbf{H}^S$
Scaled Dot Product
$\mathbf{S}^S$
Weighted Sum
$\mathbf{N}^S$
Feed Forward Network
$\mathbf{U}^S$
$\overline{\mathbf{N}}^S$
Update
g
Message Passing F
~
$\hat{Y}^S \in \mathbb{R}^{Q\times d_g}$
$\mathbf{Y}^S = [\hat{\mathbf{Y}}_1^S,...,\hat{\mathbf{Y}}_P^S]$

Fig.2. the structure of spatial Transformer

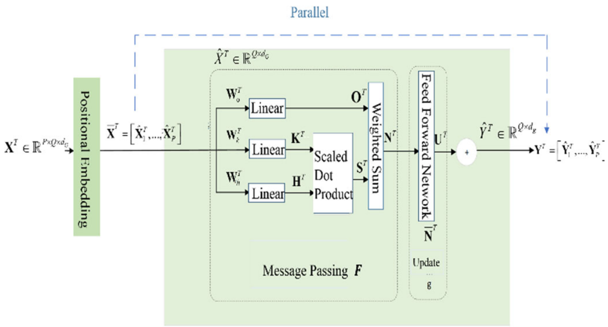

Parallel
$\hat{X}^T \in \mathbb{R}^{Q\times d_G}$
$\mathbf{X}^T \in \mathbb{R}^{P\times Q\times d_G}$
Positional Embedding
$\overline{\mathbf{X}}^T = [\hat{\mathbf{X}}_1^T, ..., \hat{\mathbf{X}}_P^T]$
$\mathbf{W}_o^T$
$\mathbf{W}_k^T$
$\mathbf{W}_h^T$
Linear
Linear
Linear
$\mathbf{O}^T$
$\mathbf{K}^T$
$\mathbf{H}^T$
Scaled Dot Product
$\mathbf{S}^T$
Weighted Sum
$\mathbf{N}^T$
Feed Forward Network
$\overline{\mathbf{N}}^T$
$\mathbf{U}^T$
+
$\hat{Y}^T \in \mathbb{R}^{Q\times d_g}$
$\mathbf{Y}^T = [\hat{\mathbf{Y}}_1^T, ..., \hat{\mathbf{Y}}_P^T]$
Message Passing F
Update
g

Fig.3. the structure of temporal Transformer

Then, $\overline{X}^{ST}$ is input into static graph convolution and dynamic graph convolution network for spatial feature learning. Since the graph convolution operation can be implemented in P time steps in parallel through tensor operations. Therefore, consider using the two-dimensional tensor $\mathbf{X}^{ST} \in \mathbb{R}^{J\times d_G}$ in $\overline{\mathbf{X}}^{ST}$ to put in Q graph convolutions to extract spatial features in parallel.

Compared with temporal Transformer, spatial Transformer uses static image convolution and dynamic image convolution to capture spatial dependence. Static graph convolution is a graph convolution network based on the spectrum method. Because the static node features are obtained by aggregating the information of its neighboring nodes according to the learned weights and predefined graphs. Use graph convolution based on Chebyshev polynomial approximation to learn structure-aware node features to capture static spatial dependence from fixed road topology. In this paper, the road topology is defined as graph $G = (\mathbf{V}, \mathbf{E}, \mathbf{A})$, and $\mathbf{D}$ is defined as the degree matrix of graph $\mathbf{G}$.The calculation formula of $\mathbf{D}$ is as follows:

$$\mathbf{D}_{ii} = \Sigma_i \mathbf{A}_{ij} \tag{2}$$

where: $\mathbf{D}_{ii}$ is the diagonal element, $\mathbf{A}_{ij}$ is the adjacency matrix established by the Gaussian kernel using the Euclidean distance between sensors.

The normalized Laplacian matrix can be expressed as $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{I}_N$ is the identity matrix. And by obtaining the maximum eigenvalue γ_{max} of $\mathbf{L}$, the scaled Laplacian matrix of Chebyshev polynomial $\mathbf{L} = 2\mathbf{L}/\gamma_{max} - \mathbf{I}_N$ can be obtained. According to the spatial -temporal position embedding feature $\mathbf{X}^{ST}$ and Chebyshev polynomial approximate fitting graph convolution kernel to obtain the structure-aware node feature $\mathbf{X}^G \in \mathbb{R}^{Q\times d_G}$, the calculation formula is as follows:

$$\mathbf{X}_j{}^G = \sum_{i=1}^{d_G} \sum_{f}^{F} \lambda_{ij,f} T_f(\mathbf{L}) \mathbf{X}_i{}^G \tag{3}$$

Where: $\mathbf{X}_j{}^G$ is the j-th channel of $\mathbf{X}^G$, $\lambda_{ij,f}$ is the learning weight, and F is the time step.

Since G is constructed based on the physical connectivity and distance between sensors, the static spatial dependence determined by the road topology can be clearly captured by the static graph convolutional layer.

However, the hot spot location of traffic flow changes over time, and static image convolution cannot capture this feature. The dynamic graph convolution can capture the implicit spatial dependencies that change over time by training and modeling high-dimensional latent subspaces. Therefore, first use dynamic graph convolution to learn linear mapping, and project the input features of each node of the road to the high-dimensional latent subspace. And adopt a self-attention mechanism to effectively simulate the dynamic spatial dependency between nodes according to the changing graph signal. Because the predefined road topology structure cannot fully represent the dynamic spatial dependency in the traffic network, the specific structure of the self-attention mechanism is shown in Fig.4. In this paper, self-attention mechanism is used to capture temporal dependence[16],[17].

The dynamic graph convolution first requires the spatiotemporal position embedding feature $\mathbf{X}^{ST}$ to be projected into the high-dimensional latent subspace through the feedforward neural network. That is, three potential subspaces need to be trained based on $\mathbf{X}^{ST}$ for each node, including query subspace $O^{ST} \in \mathbb{R}^{Q\times d_A{}^{ST}}$, key subspace KST, and value subspace $H^{ST} \in \mathbb{R}^{Q\times d_A{}^{ST}}$, The calculation formula of subspace is as follows:

$$\mathbf{O}^{ST} = \mathbf{X}^{ST}\mathbf{W}_o{}^{ST} \tag{4}$$

$$\mathbf{K}^{ST} = \mathbf{X}^{ST}\mathbf{W}_k{}^{ST} \tag{5}$$

$$\mathbf{H}^{ST} = \mathbf{X}^{ST}\mathbf{W}_h{}^{ST} \tag{6}$$

where: $\mathbf{W}_o{}^{ST} \in \mathbb{R}^{d_G\times d_A{}^{ST}}$, $\mathbf{W}_k{}^{ST} \in \mathbb{R}^{d_G\times d_A{}^{ST}}$, $\mathbf{W}_h{}^{ST} \in \mathbb{R}^{d_G\times d_A{}^{ST}}$ represent the weight matrix of $\mathbf{O}^{ST}$, $\mathbf{K}^{ST}$, $\mathbf{H}^{ST}$.

Then based on the dot product of $\mathbf{O}^{ST}$ and $\mathbf{K}^{ST}$, the $\mathbf{S}^{ST} \in \mathbb{R}^{Q\times Q}$ of the dynamic spatial-temporal dependencies between nodes is calculated. The calculation formula is as follows:

$$S^{ST} = \mathrm{softmax}\left(\mathbf{O}^{ST}(\mathbf{K}^{ST})^T / \sqrt{d_A{}^{ST}}\right) \tag{7}$$

where: the use of zoom point multiplication reduces calculation and storage costs. Softmax is used to standardize spatial dependencies, and the scale $\sqrt{d_A{}^{ST}}$ is used to prevent saturated connections caused by Softmax .Therefore, the node features $\mathbf{N}^{ST} = \mathbf{S}^{ST}\mathbf{H}^{ST}$ can be updated with $\mathbf{S}^{ST}$, the specific calculation formula is as follows:

$$\mathbf{N}^{ST} = \mathbf{S}^{ST}\mathbf{H}^{ST} \tag{8}$$

In addition, since this paper applies a shared three-layer feedforward neural network and nonlinear activation on each node, the prediction of the $\mathbf{N}^{ST}$ of the learned node feature can be further improved. The calculation formula is as follows:

$$\mathbf{U}^{ST} = \mathrm{ReLu}\left(\mathrm{ReLu}\left(\overline{\mathbf{N}}^{ST}\mathbf{W}_0{}^{ST}\right)\mathbf{W}_1{}^{ST}\right)\mathbf{W}_2{}^{ST} \tag{9}$$

where: $\overline{\mathbf{N}} = \mathbf{X}^{ST} + \mathbf{N}^{ST}$ uses residual connection for stable training, $\mathbf{W}_0{}^{ST}$, $\mathbf{W}_1{}^{ST}$, $\mathbf{W}_2{}^{ST}$ are the weights of the three-layer feedforward neural network, and the initial weight is Gaussian distribution with a standard deviation of 0.01.

Since the spatial dependence of static graph convolution and dynamic graph convolution learning cannot be directly fused, it is necessary to use a gating mechanism for feature fusion. The gating mechanism g is calculated by the static image convolution output $\overline{\mathbf{Y}}^{ST}$ and the dynamic image convolution layer output $\mathbf{X}^{G}$.The calculation formula is as follows:

$$\mathrm{g} = \mathrm{sigmoid}\left(f_S\left(\mathbf{Y}^{ST}\right) + f_G\left(\mathbf{X}^{G}\right)\right) \qquad (10)$$

where: $\mathbf{X}^{G}$ is the output of dynamic graph convolution, f_S and f_G are linear projections, which convert $\mathbf{Y}^{ST}$ and $\mathbf{X}^{G}$ into one-dimensional vectors, respectively, and $\mathbf{Y}^{ST}$ is the feature fusion output of $\mathbf{U}^{ST}$ and $\overline{\mathbf{N}}^{ST}$.

Subsequently, in this paper, YST and XST are weighted to obtain the output YST through the gating mechanism g. The specific calculation formula is as follows:

$$\overline{\mathbf{Y}}^{ST} = \mathrm{g}\mathbf{Y}^{ST} + (1-\mathrm{g})\mathbf{X}^{G} \qquad (11)$$

The output $\mathbf{Y}^{TS} \in \mathbb{R}^{P \times Q \times d_G}$ of the spatial-temporal Transformer collects the $\overline{\mathbf{Y}}^{TS}$ of P time steps and uses it as the input of the next spatial-temporal Transformer.

After k spatial-temporal Transformers extract the spatial-temporal features of the traffic flow, the output traffic flow sequence is used as the mapping between the input training of the convolutional neural network and the accident. The traffic mapping probability Y^F and the velocity mapping probability Y^S obtained by training are passed through the fully connected layer to obtain the final output. The formula is as follows:

$$Y = \beta_1 Y^F + \beta_2 Y^S \qquad (12)$$

where: Y is the final output, indicating whether there is an accident, β_1 , β_2 are the weights of the mapping probabilities Y^F, Y^S respectively.

IV. Analysis of results

A. Dataset

This paper uses two sets of real large-scale high-speed traffic flow data sets, and data fusion with the real high-speed traffic accident data sets, to obtain a traffic data set with accident labels. The high-speed traffic flow data set and the high-speed traffic accident data set are both from the Road Detection System of the California Department of Transportation (PeMS). In this paper, by matching the coordinates of the traffic flow data collection site with the coordinates of the traffic accident, the road section to which the traffic accident occurred is determined. Based on this, the traffic flow data is labeled with the accident to facilitate subsequent model training. Although the traffic flow data set selected in this article is a processed data set, there are still anomalous data, missing data, etc.

This article sets the threshold for the data according to the actual speed limit of each road and the highest value of each time period of the road in a long period of time, so as to clean the abnormal data in the data set. At the same time, according to the time period of the traffic data, the historical average method is used to fill and repair the missing data to prevent data problems from affecting the training of the model and the actual situation of the experiment.

The experiment used two large data sets, PeMSD8 and PeMSD4, which included 5 dimensions of sensor information, time slice, traffic flow, traffic speed and occupancy rate. Among them, PeMSD4 screened out 307 sensors on 29 expressways with a total of 56 days of data, and PeMSD8 screened out 170 sensors on 8 expressways, including a total of 62 days of data. According to the time and space information of PeMSD4 and PeMSD8, the accident data set collected the data of corresponding time and place. In order to facilitate the comparison of the model performance of different datasets, this paper selects 50-day data as the training set and 12-day data as the test set for each of the two data sets.

In order to help model training, this paper balances the accident data with the traffic flow data of PeMSD4 and PeMSD8 during the traffic accident prediction experiment, so that the ratio of accident data to non-accident data reaches 1:1, and built a new data set PeMSAD4, PeMSAD8.

B. Comparison method

In order to better illustrate the superiority of the accident prediction method used in this article, it is compared with the following existing research methods.

(1) Gated Recurrent Unit (GRU)

GRU achieves long-term dependence on information learning through gating. Compared with LSTM, GRU reduces the number of gates and reduces the computational cost and time cost.

(2) Convolution Long Short-Term Memory(ConvLSTM)

ConvLSTM is a variant of LSTM. It mainly changes the calculation method of input weights on the basis of LSTM. The image features are extracted through convolution operation weights, which can greatly reduce the time for extracting spatial features.

(3) Stacked denoised Autoencoder (SdAE)

SdAE is a deep architecture formed by stacking multiple DAEs. Its essence is a feature extractor, lacking a classification function, but it can be realized by adding Softmax on its top layer.

(4) Spatial-Temporal Graph Convolutional Networks (ST-GCN)

ST-GCN adds a GRU module on the basis of GCN, and uses GRU to extract temporal features on the basis of GCN extracting spatial features.

All experiments are implemented on the same host, the host operating system is Ubuntu 18.04, the memory is 64GB, the CPU used is Intel Xeon Silver, and the graphics card used is NVDIA Quadro M4000. The Python version is 2.7.12, the Pytorch version is v1.4.0, the Tensorflow version is v1.4.0, the initial learning rate is set to 0.001, and the batch size is 128, the size of the hidden layer and the cell state of the GRU unit in the codec is set to 64 at the same time, and the step size of the sliding time window in the Transformer is set to 12.

C. Evaluation index

This experiment uses Accuracy, Recall, and H-mean Score as evaluation indicators.

$$ACC = \frac{\sum_{\mathrm{n}=1}^{N} \mathrm{F}\left(\hat{\mathrm{y}} = \mathrm{y}\right)}{N} \tag{13}$$

$$\mathrm{Re}call = \frac{P_{Cor\_acc}}{P_{Cor}} \tag{14}$$

$$F1 = \frac{2 \times Acc \times \mathrm{Re}call}{Acc + \mathrm{Re}call} \tag{15}$$

Table 1. Results of accident prediction on PeMSAD4

Cluster	Method	Recall	Acc	F1 Score
Clustering	GRU	0.825	0.902	0.862
	ConvLSTM	0.810	0.932	0.867
	SdAE	0.815	0.885	0.849
	ST-GCN	0.895	0.871	0.882
	Ours	**0.873**	**0.941**	**0.905**
Not Clustering	GRU	0.804	0.882	0.841
	ConvLSTM	0.793	0.901	0.844
	SdAE	0.790	0.864	0.825
	ST-GCN	0.838	0.897	0.866
	Ours	**0.851**	**0.926**	**0.887**

where: ACC is the accuracy rate, Recall is the recall rate, $\hat{y}$ is the true value, y is the predicted value, P_{Cor_acc} is the sample predicted as an accident and correctly predicted, P_{Cor} is the sample of all accidents under real conditions, N is the number of test samples, $F(\bullet)$ is the judgment sentence, if true, output 1, otherwise output 0.

D. Result

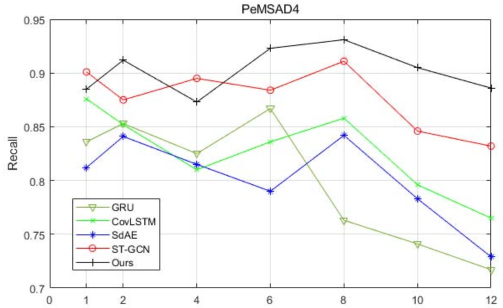
The final results of the model experiment in this paper are shown in Table **1**. This paper selects the model effects under 4 time slices for comparison, that is, the experimental results obtained by predicting the traffic data at intervals of 20 minutes to simulate the real road accident situation. In this paper, the effects of the models under 4 time slices are selected for comparison, that is, the experimental results obtained by predicting traffic data at 20-minute intervals are used to simulate the real road accident situation. It can be seen from the table that compared to SdAE that only extracts parameter features and GRU that captures time-dependent features, ST-GCN and the model used in this paper have obtained better accident prediction effects by capturing the time dependence of traffic flow and the spatial dependence of roads. Compared with ST-GCN, the model in this paper uses dynamic graph convolution to capture the hidden space dependence of road changes over time, and thus obtains a higher accuracy rate. The recall rate of the model inthis paper is less effective than ST-GCN, this is because the model captures the depth spatial-temporal dependence and the input parameters are too few, which causes some non-accident data with the spatial-temporal features to be assigned unreasonable weights and prediction errors are caused.

Among them, in the comparison of different data sets, as show in Table **2**, the model used in this article has achieved better results in PeMSAD8 with fewer nodes, this is because ST-Transformer captures a more complete depth spatial dependence for a smaller number of nodes. Compared with the model training effect using the original data, the prediction effect of the model trained using the data set divided by clustering is always better than the effect of the unclustered model. The gap under PeMSAD8 is more obvious. This is because the spatial structure of PeMSAD8 is simple, the clustering effect is better, and the model training is more effective.

Table 2. Results of accident prediction on PeMSAD8

Cluster	Method	Recall	Acc	F1 Score
Clustering	GRU	0.815	0.964	0.861
	ConvLSTM	0.906	0.912	0.889
	SdAE	0.864	0.874	0.850
	ST-GCN	0.879	0.873	0.913
	Ours	**0.898**	**0.964**	**0.930**
Not Clustering	GRU	0.763	0.915	0.832
	ConvLSTM	0.771	0.899	0.830
	SdAE	0.758	0.864	0.808
	ST-GCN	0.796	0.882	0.837
	Ours	**0.824**	**0.921**	**0.870**

The model training under different time steps also has an impact on the actual accident prediction ability of each model. This article uses the recall rate as an example to show the actual impact of the model. The specific results are shown in Figure **4**. As the time step of all models continues to increase, the model training effect will reach a peak within a range. But in the end, the training effect will continue to decline due to the time step being too large to capture the complete time dependence from the data. The model used in this article has undoubtedly achieved the best comprehensive predictive results under different time steps, and the final model training effect has decreased more than other models, this may be related to the static spatial dependence and dynamic spatial dependence captured by the model. When using the more complex PeMSAD4 dataset as input, the model used in this paper has the best prediction effect under the training of 8 time steps, and when the PeMSAD8 data set with simpler nodes is used as input, the model has the best prediction effect under the training of 6 steps. Although the prediction results of the spatiotemporal graph convolution model under smaller time step training can be better than the model used in this paper, as the time step continues to increase, the final performance lags behind the model in this paper.



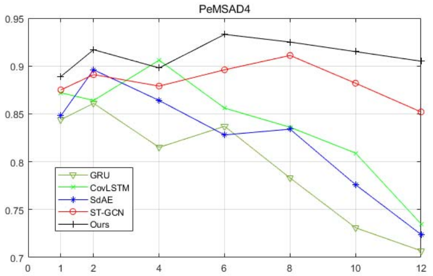


Fig.4. Comparison of results on different timestamp

The model used in this paper is processed in parallel through a sliding time window, which speeds up the training time of the model. This is demonstrated by comparing the training time with other models, as shown in Table **3**. It can be found that the ST-Transformer model used in this article is only longer than the training time of ConvLSTM and GRU, and the number of training rounds is shorter, and the model optimization is realized faster.

Compared with ST-GCN, which also captures spatial-temporal dependence, the model used in this paper captures spatial-temporal dependence while capturing deep spatial dependence and has a shorter training time.

Table 3. The comparison of model training times

Dataset	Method	Training Time(s)	Training rounds
PeMSA D4	GRU	60	35
	ConvLSTM	52	28
	SdAE	88	55
	ST-GCN	76	41
	Ours	**68**	**33**
PeMSA D8	GRU	56	32
	ConvLSTM	48	25
	SdAE	86	54
	ST-GCN	70	33
	Ours	**51**	**31**

CONCLUSION

This paper proposes a traffic accident prediction framework ST-TAP based on spatio-temporal Transformer. ST-TAP uses ClusterGAN to divide the dataset to train more targeted and proposes a spatio-temporal model based on Transformer to predict traffic accidents accurately. Experimental results from real-world traffic data show that the ST-TAP method can quickly achieve high-precision traffic accident prediction.

ACKNOWLEDGMENT

The work was partially supported by the National Natural Science Foundation of China (62073295 and 62072409), Zhejiang Provincial Natural Science Foundation (LR21F020003), Fundamental Research Funds for the Provincial Universities of Zhejiang (RF-B2020001), and Zhejiang Province Basic Public Welfare Research Project (LGG20F030008).

REFERENCES

[1] Han, X., Shen, G., Yang, X., & Kong, X. Congestion recognition for hybrid urban road systems via digraph convolutional network. Transportation Research Part C: Emerging Technologies, 121, 102877, 2020.

[2] Park, S. H., Kim, S. M., & Ha, Y. G. Highway traffic accident prediction using VDS big data analysis. The Journal of Supercomputing, 72(7), pp. 2815-2831, 2016.

[3] Zhou, X., Liang, W., She, J., Yan, Z., & Wang, K. Two-layer Federated Learning with Heterogeneous Model Aggregation for 6G Supported Internet of Vehicles. IEEE Transactions on Vehicular Technology, 2021.

[4] Kong, X., Wang, K., Hou, M., Hao, X., Shen, G., Chen, X., & Xia, F. A Federated Learning-based License Plate Recognition Scheme for 5G-enabled Internet of Vehicles. IEEE Transactions on Industrial Informatics, 2021.

[5] Ren, H., Song, Y., Wang, J., Hu, Y., & Lei, J. A deep learning approach to the citywide traffic accident risk prediction. In 2018 21st International Conference on Intelligent Transportation Systems, pp. 3346-3351, 2018.

[6] Li, Yang, and José MF Moura. Forecaster: A graph transformer for forecasting spatial and time-dependent data. arXiv preprint arXiv:1909.04019, 2019.

[7] Zhou, X., Yang, X., Ma, J., Kevin, I., & Wang, K. Energy Efficient Smart Routing Based on Link Correlation Mining for Wireless Edge Computing in IoT. IEEE Internet of Things Journal, 2021.

[8] Kong, X., Gao, H., Shen, G., Duan, G., & Das, S. K. FedVCP: A Federated-Learning-Based Cooperative Positioning Scheme for Social Internet of Vehicles. IEEE Transactions on Computational Social Systems, 2021.

[9] Shen, G., Zhao, Z., & Kong, X. GCN2CDD: a commercial district discovery framework via embedding space clustering on graph convolution networks. IEEE Transactions on Industrial Informatics, 2021.

[10] Kong, X., Wang, K., Wang, S., Wang, X., Jiang, X., Guo, Y., ... & Ni, Q. Real-time mask identification for COVID-19: an edge computing-based deep learning framework. IEEE Internet of Things Journal, 2021.

[11] Zhou, X., Xu, X., Liang, W., Zeng, Z., & Yan, Z. Deep Learning Enhanced Multi-Target Detection for End-Edge-Cloud Surveillance in Smart IoT. IEEE Internet of Things Journal, 2021.

[12] Wang, J., Kong, Y., & Fu, T. Expressway crash risk prediction using back propagation neural network: A brief investigation on safety resilience. Accident Analysis & Prevention, 124, pp. 180-192, 2019.

[13] Wenqi, L., Dongyu, L., & Menghua, Y. A model of traffic accident prediction based on convolutional neural network. In 2017 2nd IEEE International Conference on Intelligent Transportation Engineering. pp. 198-202, 2017.

[14] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G. J., & Xiong, H. Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908, 2020.

[15] Mukherjee, S., Asnani, H., Lin, E., & Kannan, S. Clustergan: Latent space clustering in generative adversarial networks. In Proceedings of the AAAI conference on artificial intelligence. Vol. 33, No. 01, pp. 4610-4617, 2019.

[16] Yuan, Z., Zhou, X., & Yang, T. Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 984-992, 2018.

[17] Zhang, J., Zheng, Y., Qi, D., Li, R., Yi, X., & Li, T. Predicting citywide crowd flows using deep spatio-temporal residual networks. Artificial Intelligence, 259, pp. 147-166, 2018.