Research on Crowd Movement Trajectory Prediction Method Based on Deep Learning

Ruikun Wang[1], Xinyu Gu[1,2(✉)], and Dongliang Li[1]

[1] Beijing University of Post and Telecommunication, 10 Xitucheng Road, Beijing 100876, China
guxinyu@bupt.edu.cn
[2] Purple Mountain Laboratories, Nanjing 211111, China

Abstract. Since the 21st century, the vigorous development of the Internet has brought about the rapid rise of social platforms. People's desire to share has been satisfied, and the information such as time and location shared by users has become the trajectory data of users. The analysis of these trajectory data is helpful to the study of crowd behavior, among which the prediction of crowd movement trajectory is an important content of trajectory data analysis.

This paper investigates the Transformer model which has an excellent performance in the field of Natural Language Processing (NLP). According to the characteristics of the data adopted in this paper, a prediction method of crowd movement trajectory based on the Transformer model is proposed and the future trajectory prediction is realized. Markov model, Long Short Term Memory Network (LSTM), and Gated recurrent unit network (GRU) are selected as baseline methods to compare with the model in this paper. The final results show that the prediction method proposed in this paper performs well in the dataset of this paper. The result also shows that the prediction methods based on deep learning have higher accuracy in predicting the future movement trajectory of the crowd compared with the prediction scheme based on the traditional model, even other parameters.

Keywords: Trajectory prediction · Deep learning · Transformer · Neural networks

1 Introduction

In human daily life, social platforms play a rather important role, and users use them to post information and communicate with other users. Some users often choose to upload their location when posting messages on social platforms, and this information shared by users is aggregated into a huge collection of user location data. For example, white-collar workers in cities usually go to

F. Gao et al. (Eds.): ChinaCom 2022, LNICST 500, pp. 390–402, 2023.
https://doi.org/10.1007/978-3-031-34790-0_30

work during the day and go home at night to rest. Students' locations usually change back and forth between school and home. Cab drivers' locations change irregularly throughout the city. By studying the movement patterns of humans in the real world, some reasonable inferences can be made about the content of human activities, which in turn can rationally explain various patterns of human behavior.

At present, the analysis of crowd movement patterns mainly faces demand from both academic and application aspects. From the academic aspect, the development of human society cannot be separated from the various activities carried out by human beings. Moving between various locations is an important part of human activities. An in-depth understanding of crowd movement patterns can only play a role in promoting the understanding of human activities. From the application aspect, with the deepening of city intelligence, personalized tertiary industries are also gradually appearing on the historical stage. Such as user location-based network service platforms, which have also developed rapidly in recent years and generated great market benefits. Their development path is to provide increasingly differentiated services for different groups of users and to provide customized service content according to each user. It is necessary to study the movement patterns of their service targets.

2 Previous Work

The research on the crowd movement trajectory prediction method is an important branch of crowd movement patterns research. By processing historical trajectory data and building a data model composed of various parameters, it can achieve the purpose of predicting the user's future arrival location. In this process, people's ideas about this direction are changing. For example, Gambs [1] built a Mobile Markov chain (MMC) model for the crowd's movement trajectory. Then, many prediction models based on the Markov model have emerged, such as the Hidden Markov model (HMM) proposed by Mathew [2]. In addition, many other traditional prediction methods based on parametric models have emerged one after another, such as the History average model (HA) [3], the Autoregressive integrated moving average model(ARIMA) [4], the Spatial-temporal autoregressive integrated moving average model (STARIMA) [5], and Vector autoregressive model (VAR) [6], the Decision tree model (DT) [7,8], and the Gaussian process model (GP) [9,10], etc., have all played their roles in the study of crowd movement trajectories.

With the continuous updating and iteration of the crowd movement trajectory prediction methods, the defects of various methods are gradually discovered. Methods based on traditional models perform poorly in solving nonlinear and non-smooth Spatio-temporal sequence prediction problems because traditional parametric models rely on fixed patterns for prediction, which are difficult to capture dynamic trajectory data features. Traditional models easily lose the ability to fit the data when dealing with massive trajectory data and fail to achieve the desired prediction accuracy [11]. In recent years, the research on crowd-moving

trajectories based on deep learning has made remarkable progress. A series of research results show that methods based on deep learning have obvious advantages in solving the problems of dynamic trajectory updates and long-term moving trajectory prediction. For example, recurrent neural network (RNN) [12], long short-term memory network (LSTM) [13], gated recurrent unit network (GRU) [14] and Time-convolutional network (TCN) [15], the encoder-decoder-based crowd movement trajectory prediction model proposed by Jianwei Chen [16], and the attention mechanism-based recurrent neural network [17] proposed by Jie Feng et al. have shown far better performance than traditional prediction models. RNN can extract time-series features from trajectory data. LSTM solves the gradient explosion problem of RNN. GRU simplifies the structure of the LSTM and reduces training costs. These three methods ignore spatial correlations between time sequences when applied to Spatio-temporal sequences predicting. TCN uses the Convolutional Neural Network (CNN) structure to capture spatial relationships while using RNN structure to capture temporal relationships. Chen's model uses the Graph Convolutional Neural Network (GCN) structure to capture spatial relationships and uses Bi-LSTM to solve the problem of RNN structure. In recent years, the attention mechanism was beginning to be applied to the trajectory prediction areas. In this paper, we adopt a crowd movement trajectory prediction method based on deep learning and attention mechanism to solve the previous problem.

3 Method

The Transformer model was originally proposed by the Google team in 2017 and was first applied to machine translation [18], which abandoned the traditional RNN model to extract sequence information and pioneered the attention mechanism to achieve fast parallel computation, improving the defect of slow training speed of RNN model. The standard Transformer model consists of four modules: Input, Output, Encoder, and Decoder. In this paper, a position encoding mechanism is used to enable the neural network to obtain the order of the vectors, and a mask padding mechanism is used to improve the sparsity of the input data. The Encoder module is split from the standard Transformer model as the core part of the model, and the Dropout mechanism is added to prevent the model from overfitting. The final model outputs the prediction through a fully connected layer. Our model is shown in Fig. 1.

3.1 Position Encoding

After the vectors are input into the model, considering that the Transformer model does not have a sequential unit structure like RNN, positional encoding (PE) is performed on the input vectors for the neural network to obtain the position information of the vectors, i.e., the internal ordering of the sequence. There are two ways to obtain the positional encoding, one is to learn through data training, and the other is to generate the positional encoding directly using

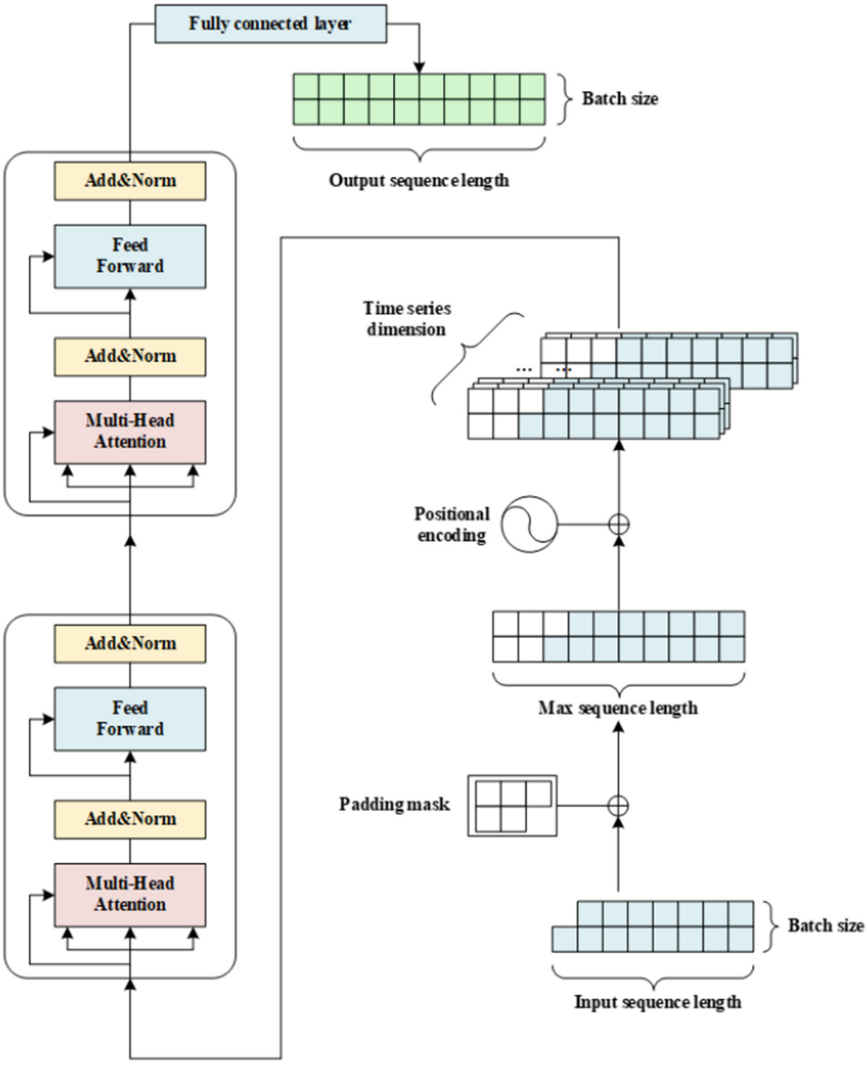

Fig. 1. Crowd movement trajectory prediction model

the formula. In this chapter, we use the positional encoding based on the sine and cosine function, which generates the corresponding encoding for each position by using the sine and cosine functions of different frequencies and adding it to the input vector of the corresponding position. The dimensionality of the position encoding vector must be the same as the dimensionality of the input vector during this calculation. The equation for calculating the position encoding based on the sine and cosine functions is shown in Eq. 1.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right)$$
$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{\text{model}}}}}\right) \quad (1)$$

where pos denotes the absolute position of a single feature in the vector, $pos = 0, 1, 2\ldots$, d_{model} denotes the dimensionality of the input vector, The dimensionality of the input vector is often large, so if the input vector and the position code are spliced, it will increase the training difficulty and the training time, so it is often chosen to add the input vector and the position code, but if the dimensionality of the input vector is small, the splicing method can also be used.

3.2 Padding Mask

The padding mask is used in Natural language procession (NLP) to deal with indeterminate length sequences of information, and a set of indeterminate length sequences of data for training will have the problem of misalignment, at this time, the short sequences will naturally fill in the length so that the data become neat, this is the padding (padding) of This is the idea of padding. For the crowd track data in this paper, because the time interval of each user's punch card is not fixed, so the whole data set becomes very sparse after the fixed time interval interception, so it is necessary to fill the gaps, and the gaps in the data sequence are filled to a negative infinite number in this paper. In order to eliminate the negative impact of useless information, it is necessary to change the weight of the padded part to a number close to 0 (mask) when passing through the softmax layer, so that the useless elements are masked to the maximum extent, the combination of these two mechanisms is the mask padding, for the Encoder part, it is only necessary to use the Padding mask for processing, while for the Decoder part, it is also necessary to Subsequent mask is used to prevent the test data from being read by the neural network, this paper uses five-fold cross-validation to replace this part of the function, so this chapter does not make more introduction.

3.3 Encoder Part

The Encoder module in the Transformer model structure consists of Multi-head attention and Feedforward neural network (FNN), which is used as the core part of the model in this chapter, shown in Fig. 2.

Attention, Self-attention and Multi-head Attention. The predecessors of Multi-head attention are attention and Self-attention. Attention is a mechanism designed to mimic human attention, shown in Fig. 3: The core attention mechanism is a weighted sum of the Value values of the elements in the Source, and the

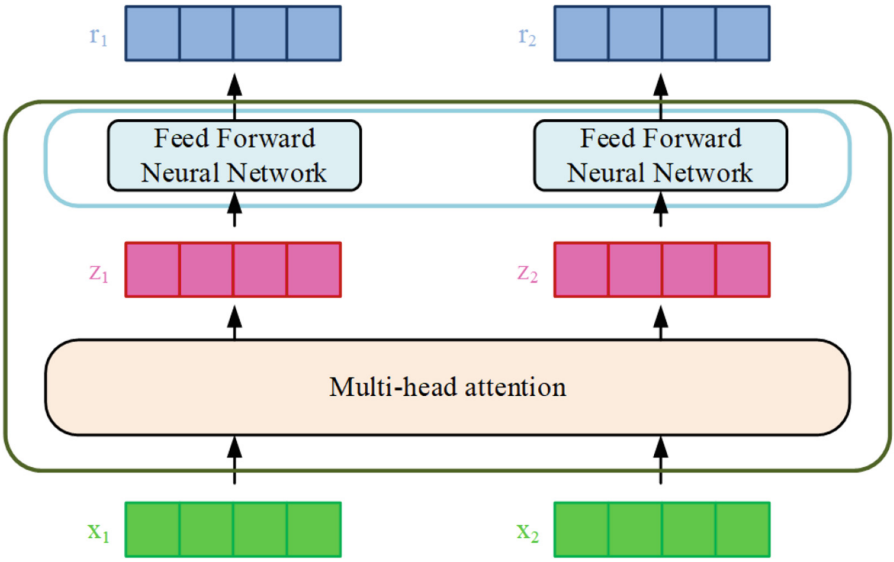

Fig. 2. Encoder part structure

Query and Key are used to calculate the weight coefficients of the corresponding Value, shown in Eq. 2:

$$Attention(Query, Source) = \sum_{i=1}^{L_x} Similarity(Query, Key_i) * Value_i \quad (2)$$

The calculation of Similarity in Eq. is to directly calculate the dot product of Query and Key, take this result as the similarity result of Query and Key, and later introduce the calculation method similar to SoftMax to numerically convert the similarity calculation result. The purpose of such calculation is twofold: on the one hand, the result can be normalized, and the original result can be organized into a probability distribution with the sum of all On the other hand, it is also possible to make the weights of relatively important elements larger through the inherent mechanism of SoftMax function, which makes them more prominent and helps to focus the attention of the network. After the similarity calculation, the weight coefficients are weighted and summed to obtain the attention coefficient.

The advantage of the attention mechanism is that it is fast because it no longer relies on RNN-based sequential decoders, solves the problem that RNNs cannot be computed in parallel, and can capture key information with excellent local performance. The disadvantage is that information is not available between medium and long distances. To improve this disadvantage, the Self-attention mechanism has emerged. In the attention mechanism, all computations occur between Source and Target, and the Self-attention mechanism makes a change by limiting the computation to the Source or Target, solving the problem of medium

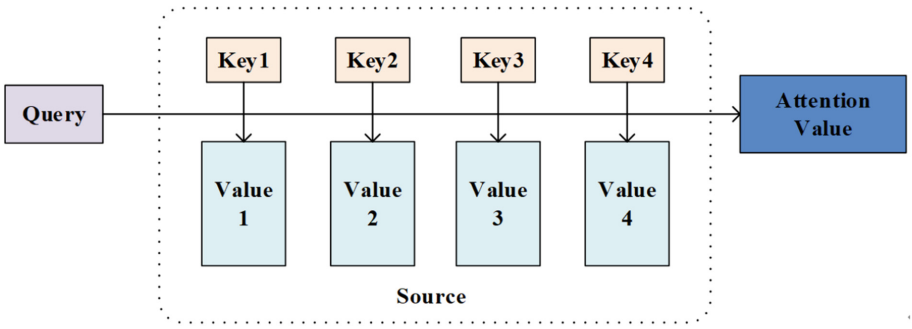

Fig. 3. Attention mechanism structure

and long-distance information capture. Both attention and Self-attention are calculated by multiplying the input vector with the weight matrix to transform the output vector. The weight matrixes W^Q, W^K, and W^V are all the same. The advantage of Multi-head attention is that it uses different weight matrixes in the calculation. It allows for a richer hierarchy of attention, allowing for multiple perspectives on the data being trained.

Feedforward Neural Network. Multi-head attention takes the position encoded data through different weight matrices, calculates multiple sets of output results, uses splicing processing to obtain a larger Z matrix as the output, and then inputs Z into the feedforward neural network, shown in Eq. 3

$$FFN(x) = \max\left(0, xW_1 + b_1\right) W_2 + b_2 \tag{3}$$

where x denotes the output Z of Multi-Head attention. The fully connected layer here uses a two-layer neural network, which first undergoes a linear transformation. Then a nonlinear transformation through the ReLU function, and finally another linear transformation. The purpose of the feedforward layer is to map the input Z to a higher dimensional space and then filter it by the nonlinear function ReLU. Finally, change Z back to the original dimension after the filtering.

Add&Normalize Part and Dropout. The final layer of the Encoder part is the Add&Normalize layer. Add means let the output vector be put into a Residual Neural Network (Res-net). In deep learning, we often encounter the problem of network degradation, which refers to the phenomenon that the network's Loss tends to stabilize at the beginning of training as the number of layers deepens, and then increases at the end of training when the number of layers continues to deepen, and the existence of residual blocks is to solve this problem. We choose Layer normalization in our model. Layer normalization is to calculate the mean and variance of all vectors in each layer and then normalize them to 0–1. The

advantage of Layer normalization is that it is not affected by different Batch-size. The output of this layer will be input to the next Encoder layer.

The dropout mechanism is to turn off some neurons to make some limitations on the fitting ability of the neural network. Let the neurons in the hidden layer stop working with a certain probability p during the forward propagation, which will lead to the simplification of the network as well as random changes. The network will not be dependent on the local features, which can make the model more generalizable, as shown in Fig. 4. In the process of Dropout, each time the hidden neurons that are turned off are different, so it is equivalent to training different neural networks. So the training results are different. The result is more like training different neural networks and averaging them, which can offset some of the "opposite" fitting results, thus preventing overfitting overall. Moreover, Dropout forces the neural network to discard fixed implicit relations and learn more robust features instead, reducing the possibility that the neural network is sensitive to specific data, which is why the Dropout mechanism can prevent overfitting. Finally, just as the emergence of gender in biology allows species to reproduce offspring that are adapted to their environment, Dropout effectively prevents the effects of overfitting on the model through a similar mechanism.

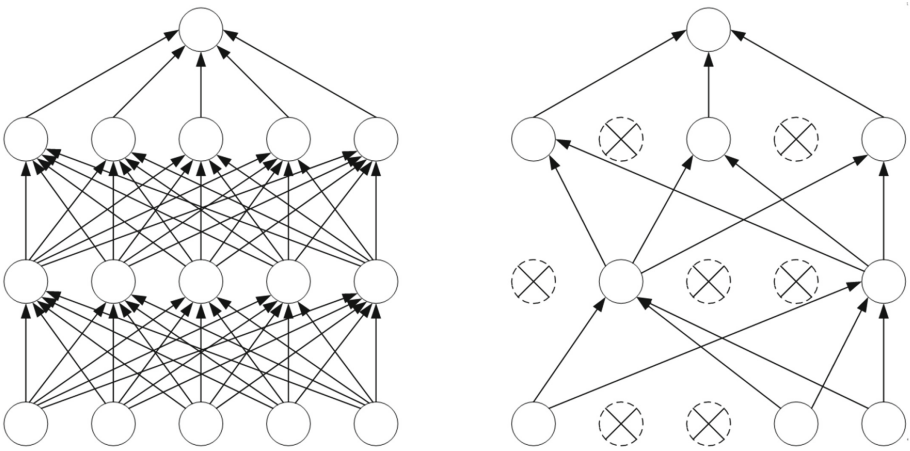

Fig. 4. The principle of the Dropout mechanism

4 Experiments

1) *Dataset*: The dataset [19] used in this paper is derived from user check-in data from the Foursquare website in Tokyo, Japan over a considerable period of time, including location information shared by users who visited various event venues to clock in over a ten-month period. As is shown in Table 1:

Table 1. Dataset

Dataset	Number of users	Check-in times	City	Duration	Number of venues
TSMC-Tky	2293	573703	Tokyo	10 months	61858

2) *Parameters*: Due to the sparsity of the trajectory sequence data, in this paper, the sequence is first complemented by the Padding Mask. Then enter Multi-head attention layer, where the number of attention heads is set to 8. The feature dimension is set to 512. The number of Encoder Layers is set to 2. The end part of the Encoder layer is Layer normalization, which is used to speed up training and improve training stability. Finally, a fully-connected layer is added to project the tensor into the form $[Batch-size, output-len]$. In this paper, we choose StepLR for the dynamic update of the learning rate. The initial learning rate is set to 0.00001, γ is set to 0.96, and step-size is set to 3, i.e., the learning rate is updated every 3 epochs. All parameters are shown in Table 2:

Table 2. Parameters of our model

Parameters	Values
input dimension	100
epoch	150
Batch-size	50
feature-size	512
Encoder-layer	2
n-head	8
Initial-learning rate	1e−5
Dropout	0.5

3) *Metrics*: The loss function used in this paper is the mean squared error function (MSE), shown in Eq. 4.

$$MSE = \frac{1}{n}\sum_{i=1}^{m} w_i \left(y_i - \hat{y}_i\right)^2 \tag{4}$$

Since the trajectory data used in this paper are continuous rather than discrete, the problem under study is a regression problem. In order to compare with the Markov model, the problem is converted to a classification problem. We choose $top@k$, $Recall$ and $F1-score$ as model performance metrics. The calculation formula of them are shown in Eq. 5, Eq. 6 and Eq. 7,

$$\text{top}@k = \frac{1}{|u|}\sum_{|u|}\sum_{j}^{|l_u^*|} \frac{\left|l_{u,j}^* \cap S_{u,j}^k\right|}{|l_u^*|} \tag{5}$$

$$\text{Recall}\,@k = \frac{1}{|u|}\sum_{|u|}\sum_{j}^{|l_u^*|}\frac{\left|S_{u,j}^{k} \cap S_{u,j}^{visited}\right|}{\left|S_{u,j}^{visited}\right|}$$
$$\text{Precision}\,@k = \frac{1}{|u|}\sum_{|u|}\sum_{j}^{|l_u^*|}\frac{\left|S_{u,j}^{k} \cap S_{u,j}^{visited}\right|}{k} \tag{6}$$

$$F_1 - score = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{7}$$

where $l^*(u,j)$ denotes the actual venue, and $S_k(u,j)$ denotes the set of the top k candidate predictions with the highest probability. In our Experiment, we choose $k = 1, 5$. $S_{u,j}^{\text{visited}}$.

4) *Baselines*: We compare our work with following several works based on generative models, including Markov, LSTM and GRU.

5 Disscusion

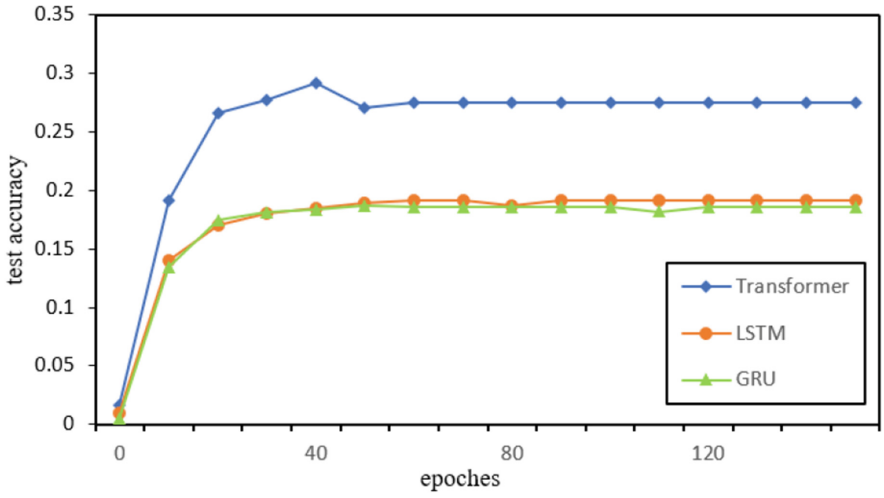

Fig. 5. Curves of test accuracy

It can be seen in Fig. 5 that the accuracy of our model performs best among all works. And as is shown in Fig. 6, the prediction accuracy of our model on the experimental dataset is 8.4% better than LSTM and 8.9% better than GRU. The Markov model cannot capture the non-Markovian properties in long-term trajectory data, such as periodicity, etc. The performance difference between the LSTM model and the GRU model is small, and both of them are greatly improved compared to the Markov model because they solve the information

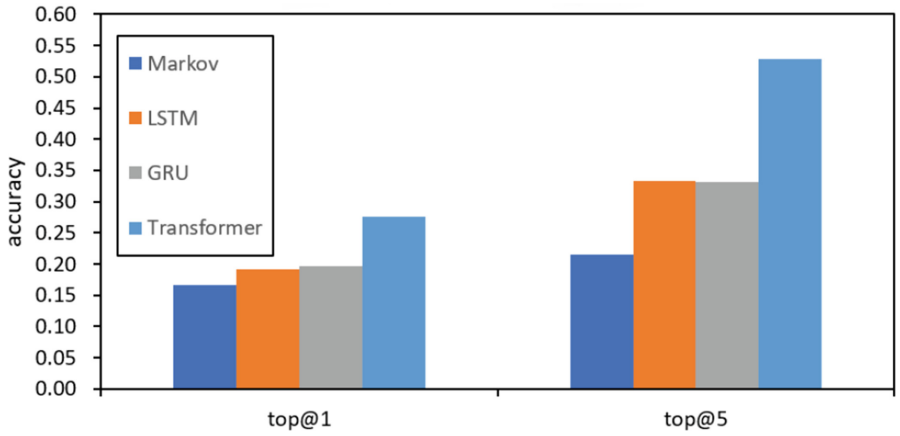

Fig. 6. Performance in TSMC_Tky

acquisition problem for long trajectories. In addition, Recall and F1-score of several models were compared in Table 3. We demonstrate the superiority of crowd movement trajectory prediction schemes based on deep learning methods over those based on traditional parametric models from another perspective, as well as the superiority of the Transformer model when dealing with sequential data.

Table 3. Recall and F1-score of all models

Model	Recall@1	Recall@5	F1-score@1	F1-score@5
Markov	0.091	0.132	0.091	0.147
LSTM	0.191	0.285	0.191	0.255
GRU	0.186	0.273	0.186	0.283
Our model	**0.275**	**0.401**	**0.275**	**0.434**

6 Conclusion

In this paper, the latest Transformer model is introduced to implement crowd movement trajectory prediction. The position encoding enables the model to obtain the position relationship of vectors. The sparsity of data is improved by the mask-filling mechanism. Attention and its variants Self-attention and Multi-head attention are introduced respectively, and the Encoder layer based on Multi-head attention and FNN is taken as the core of the model in this chapter, followed by adding residual blocks for preventing neural network degradation. We use a normalization mechanism to improve training stability as well as speed

up training. We add a Dropout mechanism to prevent overfitting of the model. Several experiments on real datasets have demonstrated the effectiveness of the model. The model used in this paper has a relatively small number of network layers due to the small amount of data. For massive amounts of data, a deeper network structure can help capture more accurate crowd movement patterns.

Acknowledgment. This work was supported in part by the State Major Science and Technology Special Projects (Grant No. 2018ZX03001024)and in part by the National Key Research and Development Program (Grant No. 2022YFF0610303).

References

1. Gambs, S., Killijian, M.-O., del Prado Cortez, M.N.: Next place prediction using mobility Markov chains (2012)
2. Mathew, W., Raposo, R., Martins, B.: Predicting future locations with hidden Markov models. In: Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp 2012, pp. 911–918. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2370216.2370421
3. Tikunov, D., Nishimura, T.: Traffic prediction for mobile network using holt-winter's exponential smoothing. In: 2007 15th International Conference on Software, Telecommunications and Computer Networks, pp. 1–5. IEEE (2007)
4. Xu, C., Xiang, W., Ji, M., et al.: Hybird forecasting model based on ARIMA and self-adaptive filtering. Comput. Appl. Softw. **35**(11), 302–306 (2018). (in Chinese)
5. Duan, P., Mao, G., Yue, W., Wang, S.: A unified STARIMA based model for short-term traffic flow prediction. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 1652–1657. IEEE (2018)
6. Wu, L., Zeng, X., Zhou, W.: A prediction method of interval series based on vector autoregression a multiple linear regression. Guilin University of Electronic Technology (2021). (in Chinese)
7. Song, Y.-Y., Ying, L.: Decision tree methods: applications for classification and prediction. Shanghai Arch. Psychiatry **27**(2), 130 (2015)
8. Ma, K.: Spatiotemporal sequences prediction methods based on decision tree method. Inf. Technol. Inform. **10**, 35–37 (2017). (in Chinese)
9. Ye, W., Cai, C., Ping, Y., et al.: UKF estimation method incorporating gaussian process regression (2019)
10. Li, S., Zhou, Y., Yue, C., et al.: Application of gaussian process mixture model on network traffic prediction. Comput. Eng. Appl. **56**(5), 186–193 (2020). (in Chinese)
11. Yang, H., Pan, Z., Bai, W.: Review of time series prediction methods. Comput. Sci. **46**(1), 21–28 (2019). (in Chinese)
12. Yu, J., de Antonio, A., Villalba-Mora, E.: Deep learning (CNN, RNN) applications for smart homes: a systematic review. Computers **11**(2), 26 (2022)
13. Chandra, R., Jain, A., Singh Chauhan, D.: Deep learning via LSTM models for COVID-19 infection forecasting in India. PloS One **17**(1), e0262708 (2022)
14. Zhang, N., Zhang, N., Zheng, Q., Xu, Y.-S.: Real-time prediction of shield moving trajectory during tunnelling using GRU deep neural network. Acta Geotech. **17**(4), 1167–1182 (2022)
15. Sang, H., Chen, W., Wang, H., Wang, J.: Pedestrian trajectory prediction model based on multi-model space-time interaction. Acta Electon. Sinica 1 (2022)

16. Chen, J.: Research on crowd trajectory prediction based on deep learning (2020). (in Chinese)
17. Feng, J., et al.: DeepMove: predicting human mobility with attentional recurrent networks. In: Proceedings of the 2018 World Wide Web Conference, pp. 1459–1468 (2018)
18. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472 (2017)
19. Yang, D., Zhang, D., Zheng, V.W., Yu, Z.: Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNS. IEEE Trans. Syst. Man Cybern.: Syst. **45**(1), 129–142 (2015)