UT-ATD: Universal Transformer for Anomalous Trajectory Detection by Embedding Trajectory Information

Yun Zhang*†, Nianwen Ning*†, Pengpeng Zhou*† and Bin Wu*†
*Beijing Key Laboratory of Intelligence Telecommunications Software and Multimedia, School of Computer Science
Beijing University of Posts and Telecommunications, Beijing, China
†{sigmarising, nianwenning, zhoupengpeng, wubin}@bupt.edu.cn

***Abstract*—Due to the development of the transportation industry, a large amount of trajectory data is pouring into the Internet all the time. Based on these trajectory data, anomalous trajectory detection technology provides great support for traffic safety assurance and traffic risk prediction. Most existing anomalous trajectory detection methods are based on trajectory's physical characteristics or representation learning, and they achieve good performance in a few scenarios. But they still face the following problems. (1) The imperfect utilization of trajectory points. (2) The sparsity of trajectory data, which leads to generalization issues. (3) Longer model training time consumed, which can't adapt to the large amount of trajectory data generated every day. To solve the above problems, we propose a novel anomalous trajectory detection model based on Universal Transformer, called UT-ATD. UT-ATD captures the information of trajectory positions by learning trajectory embedding for classification. UT-ATD has a faster training speed, relatively few model parameters, and sufficient portability, which are ideal for the realistic scene requirements. Our model achieves state-of-the-art performance in most aspects, and its effectiveness is verified by a series of experiments on the real-world taxi trajectory dataset.**

***Index Terms*—Anomalous Trajectory Detection, Trajectory Embedding, Universal Transformer**

I. Introduction

With the popularity of mobile smart terminals, numerous GPS trajectory information is generated in various scenarios. The trajectory information promotes the development of a large number of trajectory data mining studies, such as urban traffic navigation, urban construction planning, vehicle density monitoring, vehicle behavior prediction and other fields. Among them, anomalous trajectory detection becomes an increasingly important research direction.

Anomalous trajectory detection plays an essential role in two types of tasks. (1) To improve the service quality of taxis in the taxi field. Anomalous trajectory detection of taxis can prevent passengers from overpaying for deliberate bypasses by taxi drivers. It also allows taxi companies to respond in time and punish irresponsible drivers. (2) To detect the safety and unexpected status of transport vehicles in the field of transport tasks. For example, in remote prisoner escorts, anomalous trajectory detection can be used as an auxiliary detection method of the security system to assess the risk by determining whether the escort route meets the expectation. Anomalous trajectory detection can estimate whether the safety of valuables has been threatened in the transportation of extremely important items (such as museum relics).

DOI reference number: 10.18293/DMSVIVA2021-011

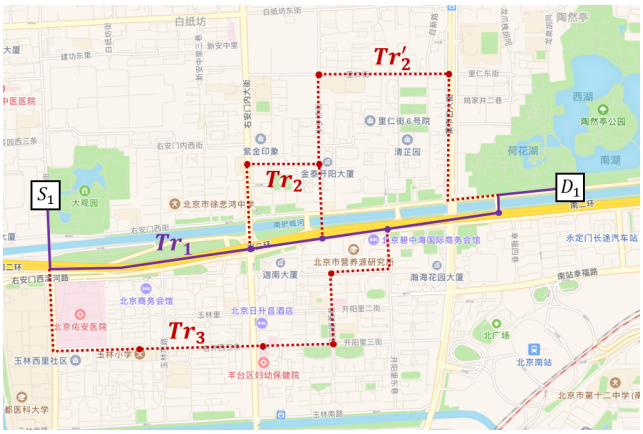


Fig. 1. The vital role of trajectory points information. Trajectory points play an important role in anomalous trajectory detection, as different anomalous trajectories can share the same anomalous trajectory points. Therefore, the trajectory points of one anomalous trajectory can make an important contribution to the detection of other anomalous trajectories.

In the real world, the anomalous trajectory detection task often needs to meet these requirements: (1) The model calculation speed is as fast as possible to fit the large amounts of trajectory data generated every moment. (2) The model should have good adaptation and generalization between different data sources. (3) The model is as portable as possible for broader deployment. In this paper, our target is to build an effective and flexible model to fit these requirements.

In the past years, there are many researches in anomalous trajectory detection, but they can't fit the real-world requirements well. The existing anomalous trajectory detection methods can be divided into two categories. The first category is based on trajectory's physical characteristics, such as density [1], direction [2], or isolation characteristics [3], [4] from the fragments of trajectories. However, these methods didn't take the importance of trajectory points into consideration. As it's shown in Fig. 1 that trajectories Tr_2 and Tr'_2 share the same red anomalous position.

The trajectory data sparsity is the main difficulty in anomalous trajectory detection. Because the number of trajectory points on the two-dimensional map are impossible to be summarized and recorded completely. With the increasing number of trajectory data, the methods based on trajectory's physical characteristics can cost too much in computing and have generalization issues. These methods based on trajectory's physical characteristics are seriously affected by the problem of trajectory data sparsity. The trajectory embedding by representation learning solve this problem to some extent.

The second category is based on trajectory embedding by representation learning. These methods [5], [6] map two-dimensional trajectories to one-dimensional sequences, and then use the models based on recurrent neural network(RNN) to encode the trajectory sequence into the low dimensional compact vectors. Compared to the first category methods, the process of trajectory embedding can make more use of the information of trajectory points. However, these methods stack to much RNN cells, which are time-consuming models during training. What's more, RNN can miss information when precessing long sequence [7]. In addition, these models usually have many parameters, which make them not portable enough and can not quickly adapt to the new trajectory data for self-training. For the above reasons, these methods can't apply well to the real world.

In this paper, we propose a novel model named **U**niversal **T**ransformer for **A**nomalous **T**rajectory **D**etection (**UT-ATD**), which is based on Universal Transformer. The model's self-attention mechanism makes the correlation degree of each trajectory point not decrease with the increase of sequence length. There are two entails substantial challenges in the real world anomalous trajectory detection. The first is the model needs to have enough effectiveness and probability in order to process the amount of trajectory data generated every day in different areas for self-training. The second is the model needs to have good generalization on the different data source to solve the trajectory data sparsity problem. To handle the above challenges, we utilize the self-circulation Universal Transformer, which uses the single Transformer block for trajectory embedding. We also change the trajectory points embedding layer with word2vec to make the model more scalable.

We carry out a series of experiments on a real-world taxi trajectory dataset, which has the highest sampling rate among all public datasets and better representation of real application scenarios. The experiments result showcase our model's better performance and generalization than other baselines.

The main contributions of this paper are listed as follows:

- We propose a novel anomalous trajectory detection model named UT-ATD, which utilizes the encoder of Universal Transformer as the core to learn the trajectory embedding.
- UT-ATD is effective and portable enough. It adapts to different data sources and has good generalization performance. The model's self-attention mechanism can encode trajectory points and trajectory sequence information better than existing methods. UT-ATD satisfies the real scenario requirements of anomalous trajectory detection task well.
- Experiments on the real-world taxi trajectory dataset verify the effectiveness of our proposed UT-ATD.

The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 defines the problem of anomalous trajectory detection. Section 4 introduces our model. Section 5 is the experiments and analysis. The summary and future work are shown in Section 6.

II. Related Work

In this section, we summary related work in two parts. Firstly, the research on anomalous trajectory detection based on trajectory's physical characteristics is summarized. Secondly, the techniques about trajectory embedding representation learning are introduced.

A. Anomalous Trajectory Detection Based on Trajectory's Physical Characteristics

The trajectory physical characteristics based detection methods mainly do calculation by classification, clustering or similarity [5], [8]. Zhu *et al.* [9], [10] mainly analyzed time-dependent outliers by the popular routes for each time interval. Lee *et al.* [1] divided the trajectories into sub-sequence and performed calculating based on distance and density. Knorr *et al.* [11] used large, multidimensional datasets based on distance to detect outliers. Li *et al.* [12] learned a model from multidimensional feature space-oriented on segmented trajectories. Kong *et al.* [13] did calculation based on the generated TS-segments. Zhang, Chen *et al.* [3], [4] used the isolation mechanism of anomalous trajectories. Zhang *et al.* [3] recorded a large number of historical trajectory and performed the adaptive iForest method for detection. Chen *et al.* [4] built an inverted index mechanism to get better retrieval performance. Zhang, Zhao *et al.* [14], [15] used the graph-based method for detection.

However, the detection methods based on distance, density, fragmentation didn't take the complete trajectory sequence information and positions context information into consideration. The calculation based on the historical database may suffer from a high time-consuming problem caused by trajectory data sparsity.

B. Trajectory Embedding By Representation Learning

Trajectory embedding is an extended field of word embedding [16]. Word Embedding is a good way to encode the information of words and sentences with a low dimension vector of fixed length. Zhao *et al.* [17] proposed a time-aware trajectory embedding model to deal with sequential information. Gao *et al.* [18] demonstrated that trajectory-user linking problem could be solved by trajectory embedding. Wu *et al.* [19] proposed a neural network algorithm based on spatial-temporal-semantic. In order to effectively capture the trajectory sequence information, many RNN based sequence models [5], [6], [20] are used for trajectory embedding training, and they achieved good results. However, for very

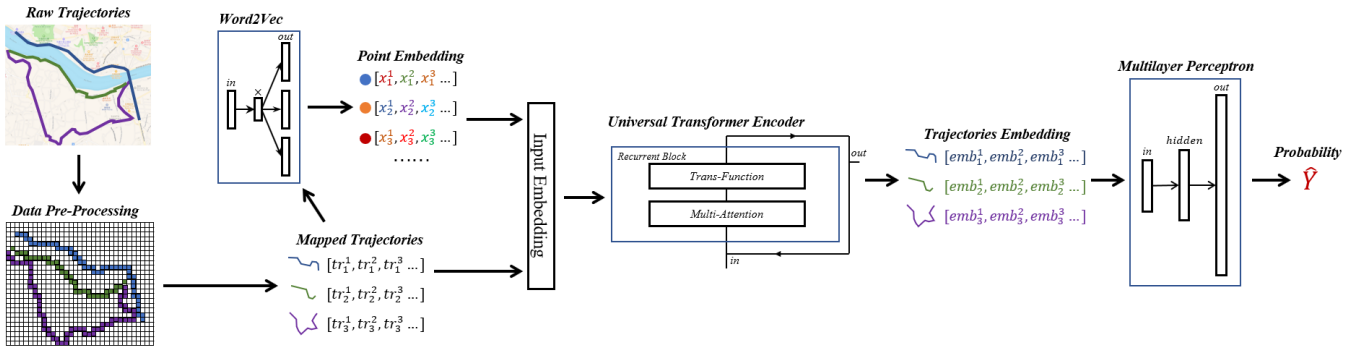


Fig. 2. The Architecture of UT-ATD. For all raw trajectories, get the mapped trajectories sequences by data pre-processing with f_{map} function. Then the word2vec(skip-gram) is used to pre-training the trajectory points embedding. Both mapped trajectories and points embedding are inputted into Universal Transformer Encoder by Input Embedding Layer. Then the output trajectory embedding is used for multilayer perceptron to calculate the probability.

long trajectory sequences, the RNN encoder cannot effectively capture all trajectory information [21]. And the training time-consuming of RNN is very high. Liu and Lane *et al.* [22] showed that the attention mechanism could improve the performance of RNN in sequence tasks. Vaswani *et al.* [7] proposed a network architecture based solely on attention mechanism named Transformer, which performed better and faster than RNN in sequence tasks. Dehghani *et al.* [23] upgraded the Transformer to Universal Transformer, which is Turing-complete and can be used in many fields such as video [24].

In order to obtain the complete trajectory information better and fully consider the role of positions in trajectory context, we use trajectory embedding by representation learning for anomalous detection. In this paper, the trajectory embedding is learned by Universal Transformer, which satisfies the real-world requirements of anomalous trajectory detection task.

III. Problem Definition

Definition 1. *Raw Trajectory Point.* The raw trajectory point tr^{pos} is the record information when GPS takes the sample, which is represented by

$$tr_i^{pos} = (id_i, time_i, lon_i, lat_i) \tag{1}$$

where (lon_i, lat_i) is the coordinate and $time_i$ is the timestamp. i denote the different raw trajectory point.

Definition 2. *Raw Trajectory.* Raw trajectory tr is a sequence of raw trajectory points collection.

$$tr_i = \left\{ tr_{i,1}^{pos} \rightarrow tr_{i,2}^{pos} \rightarrow ... \rightarrow tr_{i,len(tr_i)}^{pos} \right\} \tag{2}$$

where $tr_{i,1}^{pos}$ and $tr_{i,len(tr_i)}^{pos}$ is the source and destination of raw trajectory.

Definition 3. *Mapped Function.* For a given map, it can be divide equally into $m \times n$ grids. Every grid is represented by a unique $grid_{id}$. Mapped function f_{map} is used for convert raw trajectory point to $grid_{id}$ which it is located in.

$$f_{\text{map}}(map, m, n, tr^{pos}) = grid_{id} \tag{3}$$

where tr^{pos} is located in $grid_{id}$.

Definition 4. *Mapped Trajectory.* Mapped trajectory is a $grid_{id}$ sequence after performing mapped function on the specific raw trajectory. It's also the collection of *Mapped Trajectory point*. The mapped trajectory can be represented by equation

$$tr_i^{map} = \left\{ f_{\text{map}}(map, m, n, tr_{i,k}^{pos}) \middle| k = 1 \dots len(tr_i) \right\} \tag{4}$$

Problem Statement. *Anomalous Trajectory Detection.* For a given collection of trajectories $A = \{tr_1, tr_2, ...tr_{len(A)}\}$. Anomalous Trajectory Detection is to find out those trajectories B that are significantly different(according to the hierarchical clustering result analysis based on the similarity between trajectories) from the majority in historical datasets, where $B \subset A$.

IV. Methodology

A. Overview of UT-ATD

The architecture of UT-ATD can be found in Fig. 2. The workflow of UT-ATD is shown in Workflow 1.

The whole model can be divided into three parts: data pre-processing, trajectory embedding, and anomalous trajectory detection. In the process of data pre-processing, the two-dimensional trajectory points sequences (raw trajectory) are discrete to one-dimensional sequences (mapped trajectory). Then, we use word2vec [25] to pre-train the embedding of each mapped trajectory points. After that, the universal transformer encoder is applied to learn the embedding of mapped trajectories. Finally, a multilayer perceptron(MLP) is used to detect anomalous trajectories from trajectory embedding. In practice, the data pre-processing part is carried out separately. The Universal Transformer part and multilayer perceptron part execute joint training together.

B. Data Pre-Processing

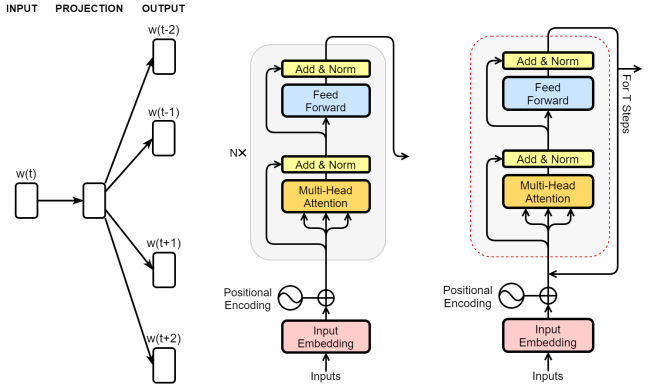
The original raw trajectory data are continuous numerical variables on the two-dimensional level. However, the number of points on the map is uncountable. The calculation cost and space cost brought by learning from raw trajectories are too expensive. The discrete step is needed to reduce the original

data's dimension while retaining the complete serialization information. In detail, the map is divided into a separate grid of $m \times n$. Through the mapping function, all points in the same grid will be given the same $grid_{id}$. The missing points in the trajectory are padded to obtain a continuous trajectory sequence by using the pre-processing method provided in [3]. We find that $100m \times 100m$ is the optimal size of the grid through repeated experiments. Both the padding and the masking techniques are applied to mapped trajectories as well. In fact, anomalous trajectories rarely appear in the historical database. In order to avoid excessive influence on the optimization of model training, we add interference data(random select the grid in sequence, and replace it with its geographic neighbor grid on the map) to each anomalous trajectory to generate some negative samples for training.

Next, the word2vec method is used to get the pre-trained embedding of each trajectory point, in order that the Universal Transformer can have a good learning effect. In practice, we choose skip-gram model for pre-training step. The model of skip-gram is shown in Fig. 3(a). The loss function is:

$$J(\theta) = -\frac{1}{T} \sum_{c=1,-m\leq j\leq m,j\neq 0}^{T} \log p(w_{c+j}|w_c;\theta) \tag{5}$$

where θ is the set of all parameters, T is the number of words in the entire corpus, m is the radius of the context window and w is the word.



(a) Skip-Gram (b) Transformer Encoder (c) Encoder of Universal Transformer

Fig. 3. Model of skip-gram and Encoder of Transformer

C. Trajectory Embedding

The encoder part of Universal Transformer is used to learn the embedding of mapped trajectory. Trajectory embedding can be learned by providing the complete mapped trajectory sequence and pre-embedding of mapped trajectory points to the encoder. The information of the trajectory points will be fully utilized in the process of calculating the trajectory embedding.

1) Transformer: Transformer is inspired by the attention mechanism. It completely replaces the RNN structure with multi-head attention and feed forward networks, which are faster and more effective than RNN based approach [7]. Transformer consists of an encoder part and a decoder part. In this paper, the encoder part is used to learn the mapped trajectory sequences and key position in trajectory context. The structure and calculation process of Transformer's encoder is shown in Fig. 3(b). Each part of this encoder will be explained below.

Scaled dot product attention. This is the main calculation logic in multi-head attention. The input consists of queries and keys of dimension d_k, and values of dimension d_v. The queries, keys, and values are packed separately as matrices Q, K and V. The calculation equation is shown below:

$$\text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

Multi-head attention. This mechanism helps model learn sequence information better. And the calculation is as follows.

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(head_1, ..., head_h)W^O \\ \text{where } head_i &= \text{Attn}(QW_i^Q, KW_i^k, VW_i^V) \end{aligned} \tag{7}$$

where h is the number of multi-head, d_{model} is the outputs dimension. The projections are parameter matrices $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ and $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$.

Feed-forward network. The representation is:

$$\text{FFN}(x) = W_2 \cdot \text{ReLU}(W_1 \cdot x + b_1) + b_2 \tag{8}$$

where W_1, W_2, b_1, b_2 are parameters of this feed-forward network. And x is the input of the network.

Positional encoding. Transformer adds the positional encoding function to get the position information of tokens.

$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{model}}) \tag{9}$$

$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}}) \tag{10}$$

where pos is the position of the current word in the sentence, i is the index of each value in the vector. Sine encoding is used in the even position, and cosine encoding is used in the odd position.

Add&Norm. The residual connection is used to ameliorate the problem of gradient disappearance in the model [26]. The output of input x after normalization is $\text{LayerNorm}(x + \text{SubLayer}(x))$, where $\text{SubLayer}(x)$ is the function implemented by sub-layer.

2) Universal Transformer: Universal Transformer is the updated version of Transformer. It uses recursive function to allow the number of the layers of Transformer can vary at will. Compared with vanilla Transformer, Universal Transformer is based on the self-circulation mechanism, which has lower model complexity, better generalization ability, lower cost of parameters, more portability and mobility. Its encoder is shown

in Fig. 3(c). An Adaptive Computation Time (ACT) [27] based dynamic halting algorithm is used to optimize computing speed.

After the calculation of Universal Transformer, the output is stacked to form the mapped trajectory embedding (refer to *trajectory embedding* in following paper). All points in trajectory are involved in the calculation of trajectory embedding.

$$Emb_i = \text{Concat}(Y_{i,1}, Y_{i,2}, ...Y_{i,MTL}) \tag{11}$$

$$Y_{i,j} = [y_{i,j}^1, y_{i,j}^2, ..., y_{i,j}^{d_{model}}] \tag{12}$$

where Emb_i is the embedding of ith mapped trajectory, and MTL is the maximum trajectory length in dataset. $Y_{i,j}$ is the output of the jth point in ith mapped trajectory in Universal Transformer.

D. Anomalous Detection

In order to detect anomalous trajectory, a MLP for classification is designed.

$$\begin{aligned} y_i =& \sigma(W_2 \cdot \sigma(W_1 \cdot Emb_i + b_1) + b_2) \\ J_{BCE} =& -\frac{1}{n}\sum_{i}^{n}(y_i^{'} \log y_i + (1 - y_i^{'}) \log(1 - y_i)) \end{aligned} \tag{13}$$

where W_1, W_2, b_1, b_2 is parameters of MLP, n is the number of samples, y is the output of MLP, and $y^{'}$ is the ground truth. For a given dataset $A = \left\{tr_1, tr_2, ...tr_{len(A)}\right\}$, the Binary Cross Entropy(BCE) loss function can be minimized to conduct joint training for the model. Dropout techniques [28] is used here to avoid over fitting problem.

E. The Workflow of UT-ATD

Workflow 1 Our UT-ATD

Input: Raw trajectory collection A

Output: The trained model UT-ATD

1: For each raw trajectory $tr_i \in A$, get the mapped trajectory set $A^{map} = \left\{tr_i^{map} \middle| i \in [0..\text{len}(A)]\right\}$ after the calculation of f_{map}. Then pad the missing points for each tr_i^{map}.

2: Get the pre-embedding matrix Emb_{pre}^{point} of each mapped trajectory point by training word2vec on A^{map}.

3: Get the maximum length K of mapped trajectories in A^{map}. Then append the 0 mark to each tr_i^{map} until its length is K. And the same length mask sequence $Mask_i$ is used to record the 0 mark information of tr_i^{map}. $Mask = \left\{Mask_i ... Mask_{\text{len}(A)}\right\}$.

4: Input Emb_{pre}^{point}, A^{map}, and $Mask$ to Universal Transformer Encoder. For each tr_i^{map}, the output is trajectory embedding Emb_i.

5: The MLP obtain Emb_i as input. The output is the probability of whether tr_i is anomalous.

Note. The Universal Transformer Encoder and MLP execute joint training.

V. Experiments

A. Evaluation

To evaluate the performance of each model, Accuracy(Acc), F_1 are used as performance evaluation criteria. The evaluations are all carried out on trajectories level.

B. Dataset

We carried out experiments on the read-world taxi trajectory dataset, which is collected from 442 taxis in Porto, Portugal from Jan. 07, 2013 to Jun. 30, 2014. The average sampling rate of GPS is $15s/point$, which is the highest sampling rate among all public datasets. This dataset can better reflect the real scene of anomalous trajectory detection. We extract five source-destination pairs (refer to *sd-pair*, *sdPair* or *sdp*) with sufficient historical trajectories data for training and testing.

Existing works tend to label the outlier manually, but the solution from [29] provided a good way to do this automatically. This methods adopts a complete-linkage clustering algorithm to hierarchically cluster the trajectories, as for outliers are "few" and "different". This method can effectively annotate the dataset. But this automatic annotating method takes a week long to extract 5 sd-pairs, which is time-consumed expensive and unbearable in real-world application as for large amount of new trajectory data is generated every day. Therefore, we only use it as a method to annotate data. The information of dataset is shown in Table I.

TABLE I
THE INFORMATION OF DATASET

dataset	sdp1	sdp2	sdp3	sdp4	sdp5
trajectories	1233	765	617	1379	4973
anomalous	54	28	37	44	270
avg trajectory len	32	31	61	51	67
max trajectory len	95	74	187	256	321
min trajectory len	17	13	42	35	40
trainset	1150	693	537	1247	4565
trainset anomalous	29	16	21	22	132
testset	83	72	80	132	408
testset anomalous	25	12	16	22	138

C. Baselines

We compare our method with the below baselines.

LCS [30]: The Longest Common Sub-sequence mechanism is a widely used method for measuring trajectory similarity. We implement LCS by comparing all trajectories in training set for every given testing set.

XGBoost [31]: This is an efficient, flexible and portable model based on gradient enhancement decision tree.

TOP-EYE [2]: TOP-EYE uses a decay function to mitigate the influences of historical trajectories, which is based on density and moving direction. We conduct this method by counting the density of each grid and compute abnormal score for test data.

LoTAD [13]: LoTAD consists of TS-segments creation and anomaly index computation. The anomaly index is computed through the density of each trajectory points.

iBOAT [4]: Anomalous trajectories will be isolated from the majority of historical dataset. iBOAT uses the inverted index mechanism to fast retrieve the relevant trajectories.

ATD-RNN [5]: This is our main compared target. In practice, it has ATD-LSTM and ATD-GRU. We test both of them in single sd-pair test. In other tests, we use ATD-LSTM as its default.

D. Results and Analysis

1) Single Source-destination Pair: The proposed model is implemented by pytorch, and set transformer's d_{model} to 64, number of multi-head-attention to 4, transformer layers to 8. The MLP's dimension is 128 and the dropout probability is 0.5. AdamW [32] is used to optimize the model.

We run our model and above baselines on different sd-pairs, the results are shown in Table II.

TABLE II
THE RESULT ON SINGLE SD-PAIR

dataset		sdPair1	sdPair2	sdPair3	sdPair4	sdPair5
LCS	Acc	0.8434	0.9444	0.9625	0.9242	0.7819
	F_1	0.7797	0.8182	0.9032	0.7059	0.5340
XGB	Acc	0.8795	0.8810	0.8375	0.9470	0.7574
	F_1	0.7619	0.4444	0.4348	0.8205	0.4469
TOPEYE	Acc	0.9629	0.9444	0.9625	0.9470	0.8431
	F_1	0.9230	0.8571	0.9032	0.8444	0.8118
LoTAD	Acc	0.6747	0.7639	0.8125	0.8333	0.6593
	F_1	0.8029	0.8661	0.8951	0.9091	0.7935
iBOAT	Acc	0.9506	**0.9583**	0.9625	0.9394	0.8750
	F_1	0.9167	0.8696	0.8276	0.7778	0.8198
ATD-LSTM	Acc	0.9518	**0.9583**	**0.9875**	0.9848	0.9020
	F1	0.9167	0.8696	0.9697	0.9545	0.8374
ATD-GRU	Acc	0.9638	**0.9583**	0.9750	**0.9924**	0.9167
	F1	0.9412	0.8800	0.9412	0.9767	0.8722
UT-ATD	Acc	**0.9880**	**0.9583**	0.9750	**0.9924**	**0.9191**
	F1	**0.9915**	**0.9752**	**0.9841**	**0.9954**	**0.9430**

The relatively low Acc scores of LCS and XGBoost may due to the fact that these methods only consider the shape information of the trajectory and ignore the historical sequences information of trajectories. The LoTAD's result low may because it lacks the information of complete sequence. TOP-EYE and iBOAT achieve considerable performance may because they use historical trajectories data for calculation. iBOAT out-performs TOP-EYE in most circumstances may be that iBOAT not only takes the similarity into consideration but also makes use of the local sequential information.

It can be seen that UT-ATD achieves the highest accuracy on most sd-pairs and gets the best F_1 on all sd-pairs. It proves that UT-ATD is superior to ATD-RNN in its ability to obtain information from trajectory. It also shows that the information of different locations in trajectory context plays a very important role in trajectory embedding. One of the reasons why on sd-pair3 have the minimum amount of data, and UT-ATD is not trained enough. But in contrast, UT-ATD achieves a better F_1 on sd-pair3 than ATD-RNN.

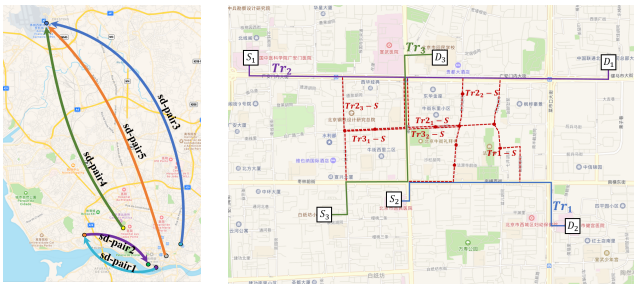
TABLE III
THE RESULT ON MULTI-SD-PAIR

dataset	ATD-RNN		UT-ATD	
	Acc	F1	Acc	F1
sdPair3+4	0.9340	0.9611	**0.9528**	**0.9714**
sdPair3+5	0.8996	0.9299	**0.9262**	**0.9459**
sdPair4+5	0.9148	0.9415	**0.9315**	**0.9520**
sdPair3+4+5	0.9145	0.9422	**0.9339**	**0.9550**
sdPair1+2+3+4+5	0.9123	0.9413	**0.9406**	**0.9598**

2) Multi-source-destination Pair: Through trajectory embedding, the extensibility of calculation among different sd-pairs is possible. Because the trajectory sequences between different sd-pairs can complement each other, the pressure of data sparsity can be alleviated to some extent [5]. We pack different sd-pairs together to make up multi-sd-pair. Then we compare UT-ATD with ATD-RNN on different multi-sd-pairs.

The geographic relationship of five sd-pairs can be seen in Fig. 4(a). What stands out in this figure is that sd-pair1 and sd-pair2 are close to each other. And sd-pair3, sd-pair4, and sd-pair5 are close to each other. The test results on combination of different sd-pairs are shown in Table III.

It can find out that, for geographical closed or not closed sd-pairs combination, UT-ATD's generalization ability is better than ATD-RNN. The probable reason is that the key positions in one sd-pair offer information for other sd-pairs. As shown in Fig. 4(b), the key positions(red) in $Tr3$-S will offer information for detecting $Tr1$-S and $Tr2$-S. For lots of very long trajectories, the information key positions offered will be more. Another possible reason is the RNN's poor ability to encode long trajectories, as for the trajectories in sd-pair3, sd-pair-4 and sd-pair5 are longer than others.



(a) Relationship of five SD-Pairs (b) Key Position Impact for Multi-Trajectories

Fig. 4. Validity of Word2vec and Universal Transformer

3) Training Speed Comparison: To compare the training speed of ATD-RNN and UT-ATD, we train both models with epoch(40), batch-size(32), mapped trajectory points pre-embedding dimension(64), and layers of neural network(6) on Linux server(Intel Xeon Gold 5118, 250G RAM, NVIDIA RTX 2080 Ti) and take average time for multiple runs.

The result can be seen from Fig. 5(a) that when training on sd-pair1 to sd-pair5, the training efficiency of UT-ATD is significantly faster than ATD-RNN. This is mainly because ATD-RNN must follow the sequence order as trajectory embedding

is computed. The next state can only be computed after the calculation of the previous state. This doesn't constrain UT-ATD, and it can compute all information in parallel. So UT-ATD naturally has a faster speed than RNN based method in training.

(a) Training Time Comparison (b) Number of d_{model}

(c) Number of Multi-head-attention (d) Number of layers

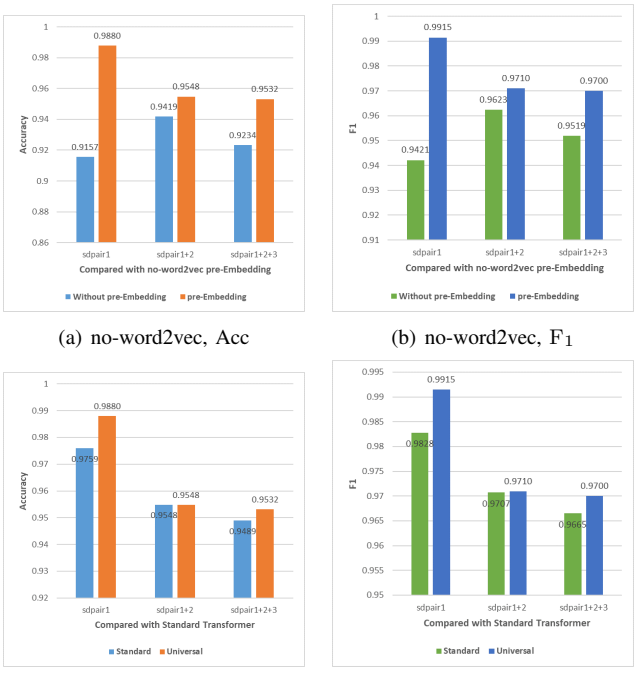
Fig. 5. Training Time and Parameter Influence

4) Parameter Experiments: To explore the influence of different parameters, we test the influence of d_{model}, number of multi-head-attention, and layers for UT-ATD on sd-pair1.

It can be seen from Fig. 5(b) that, with the increase of d_{model}, the accuracy and F_1 will increase and tend to be stable. The reason is that with the increase of trajectory embedding dimension, the represented information will increase. When $d_{model} = 64$, the information is saturated and there is no need to increase d_{model}. The Fig. 5(c) shows that accuracy and F_1 will fluctuates as num of multi-head-attention grows. One possible reason is that the length of trajectories in sd-pair1 is short and the effect of multi-head-attention to neutralize effective resolution is not obvious. Fig. 5(d) shows that the more number of layers, the lower performance will get. The reason is that the increase of layers will make model more complex, which may lead to over-fitting problem.

5) Ablation Experiments: We test the UT-ATD without word2vec pre-training (the pre-embedding is randomly initialized and adjusted during training) and compare UT-ATD to standard transformer on different scale sd-pairs.

As it's shown in Fig. 6(a) and Fig. 6(b) that the pre-training of word2vec do help the model to perform better. Fig. 6(c) and Fig. 6(d) show that Universal Transformer performs better than standard transformer, which may because the Universal Transformer is Turing-complete and more portability.



(a) no-word2vec, Acc (b) no-word2vec, F_1

(c) stand-transformer, Acc (d) stand-transformer, F_1

Fig. 6. Validity of Word2vec and Universal Transformer

VI. Conclusions and Future Work

In this paper, we propose an anomalous trajectory detection method **U**niversal **T**ransformer for **A**nomalous **T**rajectory **D**etection (**UT-ATD**). UT-ATD uses the Universal Transformer encoder to learn the embeddings of the trajectory effectively. Compared with traditional methods and RNN-based model, UT-ATD not only captures the information of complete trajectory sequence better but also considers the importance of positions in trajectory context. UT-ATD is not limited by the given source-destination pairs. It performs well in multiple combined source-destination pairs. The experiments on the real-world taxi trajectories dataset demonstrate the effectiveness of UT-ATD.

In the future, we will extend the UT-ATD for online detection and study the influence of few-shot learning on our model. Whether multi-modal based method has a good effect on anomalous trajectory detection is also worth discussing.

Acknowledgment

This work is supported by National Key Research and Development Program of China (2018YFC0831500), National Natural Science Foundation of China under Grant No.61972047, and NSFC-General Technology Basic Research Joint Funds under Grant U1936220.

References

[1] J. Lee, J. Han, and X. Li, "Trajectory outlier detection: A partition-and-detect framework," in *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, Mexico*. IEEE Computer Society, 2008, pp. 140–149.

[2] Y. Ge, H. Xiong, Z. Zhou, H. T. Ozdemir, J. Yu, and K. C. Lee, "Top-eye: top-k evolving trajectory outlier detection," in *Proceedings of the 19th ACM Conference on Information and Knowledge Management, CIKM 2010, Toronto, Ontario, Canada, October 26-30, 2010*. ACM, 2010, pp. 1733–1736.

[3] D. Zhang, N. Li, Z. Zhou, C. Chen, L. Sun, and S. Li, "ibat: detecting anomalous taxi trajectories from GPS traces," in *UbiComp 2011: Ubiquitous Computing, 13th International Conference, UbiComp 2011, Beijing, China, September 17-21, 2011, Proceedings*. ACM, 2011, pp. 99–108.

[4] C. Chen, D. Zhang, P. S. Castro, N. Li, L. Sun, S. Li, and Z. Wang, "iboat: Isolation-based online anomalous trajectory detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 806–818, 2013.

[5] L. Song, R. Wang, D. Xiao, X. Han, Y. Cai, and C. Shi, "Anomalous trajectory detection using recurrent neural network," in *Advanced Data Mining and Applications - 14th International Conference, ADMA 2018, Nanjing, China, November 16-18, 2018, Proceedings*, ser. Lecture Notes in Computer Science, vol. 11323. Springer, 2018, pp. 263–277.

[6] Y. Cheng, B. Wu, L. Song, and C. Shi, "Spatial-temporal recurrent neural network for anomalous trajectories detection," in *Advanced Data Mining and Applications - 15th International Conference, ADMA 2019, Dalian, China, November 21-23, 2019, Proceedings*, ser. Lecture Notes in Computer Science, vol. 11888. Springer, 2019, pp. 565–578.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 5998–6008.

[8] A. Belhadi, Y. Djenouri, and J. C. Lin, "Comparative study on trajectory outlier detection algorithms," in *2019 International Conference on Data Mining Workshops, ICDM Workshops 2019, Beijing, China, November 8-11, 2019*. IEEE, 2019, pp. 415–423.

[9] J. Zhu, W. Jiang, A. Liu, G. Liu, and L. Zhao, "Time-dependent popular routes based trajectory outlier detection," in *Web Information Systems Engineering - WISE 2015 - 16th International Conference, Miami, FL, USA, November 1-3, 2015, Proceedings, Part I*, ser. Lecture Notes in Computer Science, vol. 9418. Springer, 2015, pp. 16–30.

[10] ——, "Effective and efficient trajectory outlier detection based on time-dependent popular route," *World Wide Web*, vol. 20, no. 1, pp. 111–134, 2017.

[11] E. M. Knorr, R. T. Ng, and V. Tucakov, "Distance-based outliers: Algorithms and applications," *VLDB J.*, vol. 8, no. 3-4, pp. 237–253, 2000.

[12] X. Li, J. Han, S. Kim, and H. Gonzalez, "ROAM: rule- and motif-based anomaly detection in massive moving object data sets," in *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*. SIAM, 2007, pp. 273–284.

[13] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[14] J. Zhang, "Smarter outlier detection and deeper understanding of large-scale taxi trip records: a case study of NYC," in *Proceedings of the ACM SIGKDD International Workshop on Urban Computing, UrbComp@KDD 2012, Beijing, China, August 12, 2012*. ACM, 2012, pp. 157–162.

[15] X. Zhao, Y. Rao, J. Cai, and W. Ma, "Abnormal trajectory detection based on a sparse subgraph," *IEEE Access*, vol. 8, pp. 29 987–30 000, 2020.

[16] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 1188–1196.

[17] W. X. Zhao, N. Zhou, A. Sun, J. Wen, J. Han, and E. Y. Chang, "A time-aware trajectory embedding model for next-location recommendation," *Knowl. Inf. Syst.*, vol. 56, no. 3, pp. 559–579, 2018.

[18] Q. Gao, F. Zhou, K. Zhang, G. Trajcevski, X. Luo, and F. Zhang, "Identifying human mobility via trajectory embeddings," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 2017, pp. 1689–1695.

[19] F. Wu, K. Fu, Y. Wang, Z. Xiao, and X. Fu, "A spatial-temporal-semantic neural network algorithm for location prediction on moving objects," *Algorithms*, vol. 10, no. 2, p. 37, 2017.

[20] Y. Liu, K. Zhao, G. Cong, and Z. Bao, "Online anomalous trajectory detection with deep generative sequence modeling," in *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*. IEEE, 2020, pp. 949–960.

[21] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why self-attention? A targeted evaluation of neural machine translation architectures," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2018, pp. 4263–4272.

[22] B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*. ISCA, 2016, pp. 685–689.

[23] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal transformers," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[24] M. Bilkhu, S. Wang, and T. Dobhal, "Attention is all you need for videos: Self-attention based video summarization using universal transformers," *CoRR*, vol. abs/1906.02792, 2019.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[26] L. J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *CoRR*, vol. abs/1607.06450, 2016.

[27] D. Fojo, V. Campos, and X. Giró-i-Nieto, "Comparing fixed and adaptive computation time for recurrent neural networks," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018.

[28] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[29] H. Wu, W. Sun, and B. Zheng, "A fast trajectory outlier detection approach via driving behavior modeling," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017, Singapore, November 06 - 10, 2017*. ACM, 2017, pp. 837–846.

[30] J. J. Ying, W. Lee, T. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011, November 1-4, 2011, Chicago, IL, USA, Proceedings*. ACM, 2011, pp. 34–43.

[31] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. ACM, 2016, pp. 785–794.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.