EasyChair Preprint

№ 672

Anomalous Trajectory Detection using Recurrent Neural Network

Li Song, Ruijia Wang, Ding Xiao, Xiaotian Han, Yanan Cai and Chuan Shi

December 7, 2018

Anomalous Trajectory Detection using Recurrent Neural Network

Li Song[1,3], Ruijia Wang[1,4], Ding Xiao[1,4], Xiaotian Han[1,3], Yanan Cai[2,5], and Chuan Shi[1,4]

[1] Beijing University of Posts and Telecommunications
[2] Big Data Center, PICC Property and Casualty Company Limited
[3] {song200626, hanxiaotian.h}@gmail.com
[4] {Vanrhyga,dxiao,shichuan}@bupt.edu.cn
[5] caiyanan@picc.com.cn

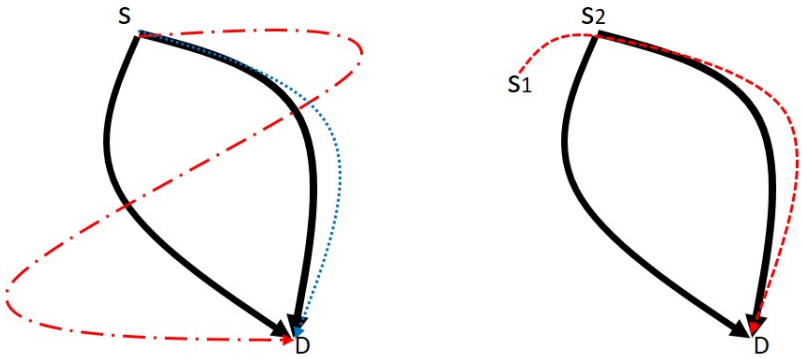
Abstract. Anomalous trajectory detection which plays an important role in taxi fraud detection and trajectory data preprocessing is a crucial task in trajectory mining fields. Traditional anomalous trajectory detection methods which utilize density and isolation approaches mainly focus on the differences of a new trajectory and the historical trajectory dataset. Although these methods can capture the particular characteristics of trajectories, they still suffer from the following two disadvantages. (1) These methods cannot capture the sequential information of the trajectory well. (2) These methods only concentrate on the given source and destination which may lead to data sparsity issues. To overcome above shortcomings, we propose a novel method called **A**nomalous **T**rajectory **D**etection using **R**ecurrent **N**eural **N**etwork (**ATD-RNN**) which characterizes the trajectory by learning the trajectory embedding. The trajectory embedding can capture the sequential information of the trajectory and depict the internal characteristics between anomalous and normal trajectory. To address the potential data sparsity problem, we enlarge the dataset between a source and a destination by taking the relevant trajectories into consideration. Extend experiments on real-world datasets validate the effectiveness of our method.

Keywords: Anomalous Trajectory Detection · Trajectory Embedding · Recurrent Neural Network.

1 Introduction

With the proliferation of global positioning system (GPS) based equipment, massive spatial trajectory data has been generated. The trajectory data represents the mobility of a diversity of moving objects and contains valuable information concerning both Service Providers (SPs) and the customers. A large number of trajectory data mining tasks [25], including map matching, trajectory compression, stay point detection, POIs recommendation, trajectory classification and anomalous trajectory detection, have been widely researched. Anomalous trajectory detection plays an important role in the fields of trajectory data mining.

For example, anomalous trajectory detection can enhance the quality of taxi services impressively, since the greedy taxi drivers who overcharge passengers by deliberately taking unnecessary detours could be detected with anomalous trajectory detection technique. Therefore, it allows the taxi companies to monitor the movements of all the taxis and to identify the dishonest drivers who tend to take routes longer than usual. A toy example of anomalous and normal trajectory between a source (S) and destination (D) (SD-Pair) can be seen in Fig. 1(a).



(a) Anomalous and normal trajectory. (b) Trajectory of two neighbor SD-Pairs.

Fig. 1. The historical trajectories between SD-pairs.

In recent years, automatically anomalous trajectory detection has attracted extensive research attention. Some existing anomalous trajectory detection methods have been proposed to deal with this problem [13, 2, 8, 23, 4, 26, 20] . [13] proposes a partition-and-detect framework for anomalous trajectory detection, which partitions a trajectory into a set of line segments and detects outlying line segments for trajectory outliers based on distance and density. IBAT [23] and the augmented version IBOAT [4] present the isolation-based anomalous trajectory detection methods which detect anomalous by comparing a new trajectory against a large collection of historical trajectories. [26] determines whether an input trajectory is an anomalous by taking spatial and temporal abnormalities into consideration simultaneously. However, most existing methods addressing the anomalous detection problem are only based on density or isolation methods. So although these methods achieve impressive performances, there still remain some limitations to these methods.

1. **Sequential Information.** These methods based on density or isolation cannot characterize the sequential information of a trajectory well. Those methods mainly focus on sub-trajectories or trajectory points and neglect the influence of sequential information of the whole trajectory.

2. **Data Sparsity.** These methods based on density or isolation also suffer from data sparsity problem. If there are N intersections in a city, then the number of SD-pairs will be up to N^2. Even worse, it is only few trajectories in historical dataset for some special intersection. As a result, it is difficult to model these SD-pairs.
3. **Computational Complexity and Space Complexity.** Traditional anomalous trajectory detection methods is time-consuming since these methods need to compute the similarity of every two trajectories in historical datasets. Moreover, for every SD-pair, a corresponding model is required.

Inspired by that word embedding [12, 17] models the co-occurrence of words by mapping the words to the low-dimensional vectors, we propose a novel method called **A**nomalous **T**rajectory **D**etection using **R**ecurrent **N**eural **N**etwork (ATD-RNN). ATD-RNN represents the trajectory by the low-dimensional vector using recurrent neural network (RNN) and detects anomalous trajectory in the embedded space. The sequential information is learned through trajectory embedding because RNN can use its internal memory to process time series data. To address the shortcoming of data sparsity, ATD-RNN attempts to train the model through the extended large amount of historical trajectories. For example, as illustrated in Fig. 1(b), there may be few historical trajectories from source S_1 to destination D. But if take the source S_2 into consideration which has sufficient historical trajectories to build a model, the situation for source S_1 could be alleviated. Since different SD-Pairs could be trained together, ATD-RNN has potential to reduce the frequency of accessing to historical trajectories and has little demand for extra space. Therefore, ATD-RNN not only solves the anomalous trajectory detection by capturing the sequential information through trajectory embedding, but also alleviates the data sparsity problem by taking different SD-Pairs into consideration. Experiments on several real-world datasets demonstrate the superior performance of the proposed method. Source code is available at Github[1].

The main contributions of this paper are listed in detail as follows:

- We present a novel anomalous trajectory detection method named ATD-RNN, which captures the sequential information of the trajectory by learning the trajectory embedding through a delicately designed recurrent neural network.
- We take different sources and destinations into consideration to alleviate the data sparsity problem. This can capture the internal characteristics from similar trajectories.
- We evaluate ADT-RNN on real-world datasets collected from 442 taxis in Porto for one year. It achieves remarkable detection results comparing to the famous anomalous trajectory detection baselines.

The rest of this paper is organized as follows. Section 2 reviews the related works. A formal definition of our problem is given in section 3. The details of

[1] https://github.com/LeeSongt/ATD-RNN

our proposed method are introduced in section 4. Experimental results of our method are compared and analyzed in section 5. And section 6 concludes this paper and outlines the future work.

2 Related Work

In this section, we give a brief review of the relevant academic literature on anomalous trajectory detection and trajectory embedding.

Anomalous trajectory detection. A considerable amount of researches have been published on anomalous trajectory detection. In general, these studies could be divided into three major categories.

The first major category is clustering, which performs automated grouping based on distance and density. [13] conducts a partition-and-detect framework for trajectory outliers detection. [11] deals with finding outliers in large, multidimensional datasets based on distance. [22] characterizes outliers as moving objects that behaved differently from the majority in trajectory streams by neighbor-based trajectory outliers definitions. However, these distance-based and density-based approaches often need to calculate the measure metrics according to the whole historical database that is time consuming.

The second category is classification, which relies on labeled training data. In order to detect anomalous efficiently and effectively, [14] constructs a multidimensional feature space oriented on segmented trajectories and then learns a model to classify trajectories as normal or abnormal. [18] proposes an anomalous behavior detection framework in a video surveillance scenario. However, these classification-based approaches require amount of labeled data which spends huge manpower and material resources.

The third category is based on patterns. [3] proposes an anomalous behavior detection framework using trajectory analysis, which includes a trajectory patterns learning module and an online abnormal detection module. [23] presents an Isolation-Based Anomalous Trajectory (iBAT) detection method which exploits the property that anomalies are susceptible to a mechanism called isolation. IBOAT [4] is an augmented version of iBAT. Specifically, in order to process trajectories online and find the anomalous at early stag, IBOAT builds an inverted index for historical trajectory data. [26] proposes a time-dependent popular routes based algorithm which takes spatial and temporal abnormalities into consideration simultaneously to improve the accuracy of the detection. [20] proposes a probabilistic model-based approach via modeling the driving behavior/preferences from the set of historical trajectories. Generally speaking, the pattern-based approaches investigate the co-occurrence of trajectory points so that the anomalous trajectory which is rarely appeared can be detected.

Trajectory Embedding. Trajectory Embedding is an extended field of word embedding [12, 17] which represents a word as a fixed length numerical vector and the co-occurrence of words can be learned from the embedded vectors. [24] proposes a time-aware trajectory embedding model in next-location recommendation systems to deal with sequential information and data sparsity prob-

lem. [19] predicts next location using a spatial-temporal-semantic neural network algorithm. Concretely, the road networks is divided into significant discrete points in terms of a distance parameter threshold and then a long short-term memory (LSTM) neural network is used to model these sequences. [7] demonstrates that trajectory-user linking problem can be solved by trajectory embedding. After the check-ins in trajectories is embedded into a low-dimensional space, the connection between a particular user and the motion patterns can be learned with a LSTM. In order to learn sequential information, trajectory embedding is applied to model the co-occurrence between the trajectory points.

3 Problem Definition

It is necessary here to clarify exactly what is meant by trajectory. After that, the formal definition of anomalous trajectory detection is introduced.

Definition 1. *(Raw Trajectory). A raw trajectory* $T = \{p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n\}$ *is a sequence of records, and each recored* p_i *is represented by* (lon_i, lat_i, t_i) *where* (lon_i, lat_i) *is a geographic coordinate and* t_i *is the timestamp.* p_1 *and* p_n *are source and destination of the trajectory, respectively.*

Definition 2. *(Mapped Trajectory). For a given* n *and* m*, a map can be split into* $n * m$ *equal sized grids. Then, a map function* $\rho(lon, lat, n, m) = grid_i$ *implements a discretization process. A mapped Trajectory corresponding to a raw trajectory* $T = \{p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n\}$ *is expressed as* $tr = \{grid_1 \rightarrow grid_2 \cdots \rightarrow grid_n\}$*, where* $grid_i = \rho(lon_i, lat_i, n, m)$*.*

Definition 3. *(Anomalous Trajectory Detection). Given a set of trajectories* $D = \{tr_1, tr_2, \cdots, tr_m\}$*, anomalous trajectory detection is to find those trajectory* R *that are significantly different from the majority in historical datasets.*

Since anomalous trajectory is rarely occurred, we want to learn the normal pattern from historical trajectories. Then, an anomalous can be detected by comparing a trajectory with these normal patterns. However, it is difficult to learn the normal patterns for a given source and destination (SD-pair). Traditional anomalous trajectory detection methods based on density and isolation mainly focus on sub-trajectory and points in the trajectory and cannot capture the sequential information of the trajectory well. In addition, these methods tend to train one model for one SD-Pair that cannot make use of similar characteristics between different SD-Pairs. Even worse, these methods concentrate on an SD-Pair may suffer from data sparsity.

In this paper, anomalous trajectory detection is conducted by learning trajectory embedding through recurrent neural network. Trajectory embedding can capture the internal characteristics between anomalous and normal trajectories.

4 Methodology

4.1 Overview of ATD-RNN

In this paper, we propose a method to detect anomalous trajectories named **A**nomalous **T**rajectory **D**etection using **R**ecurrent **N**eural **N**etwork(**ATD-RNN**). As illustrated in Fig. 2, the proposed method is mainly consists of three steps, trajectory data preprocessing, trajectory embedding and anomalous detection. In trajectory data preprocessing step, we discrete the trajectory points so that the continuous numerical variables could be fed into the trajectory embedding step. After that, a delicately designed neural network, stacked RNN, is applied to learning the trajectory embedding which can capture the sequential information and the internal characteristics of the trajectories. Then, the multilayer perceptron and a softmax layer are used to detect anomaly from the trajectory embeddings. We will describe these steps in detail in the following chapters.

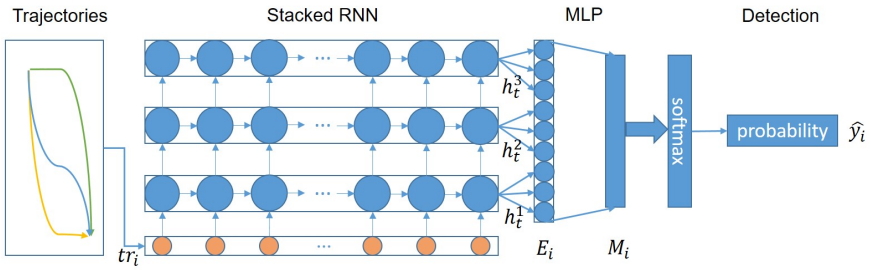


Fig. 2. The architecture of ATD-RNN.

4.2 Trajectory Data Preprocessing

In general, the raw trajectory data is continuous numerical variables. These trajectory points are enormous so that we cannot learn the trajectory embeddings for every points. Even worse, it is difficult to generalize to new points. In trajectory data preprocessing, we need a discretization step. In details, the geographic coordinate space is meshed into equal sized grids according to the hyper parameters n and m. In fact, we adjust n and m so that the size of a grid is about $100m$. Then we label each grid with a unique index. After that, the raw trajectory points located on the grid will have the same index. Although we obtain the mapped trajectories, we have to notice that the length of each trajectory data is not equivalent. Therefore, we utilize some padding operations to align the trajectories. After that, a trajectory tr_i is represented as:

$$tr_i = \{x_1, x_2, \cdots, x_m\}, \tag{1}$$

another critical problem in anomalous detection is that the anomalies are often rarely appear in historical datasets. This may be an obstacle in the process of optimization. To alleviate this problem, we sample some anomalous trajectories and make disturbances to generate new anomalous data. In practice, we randomly select some trajectory points from an anomalous trajectory and replace those points with their neighbors.

4.3 Trajectory Embedding

We use recurrent neural network to learn the sequential information of the trajectory. After the whole trajectory feeds into a stacked recurrent neural network, we can learn the trajectory embedding.

Recurrent Neural Network. Recurrent neural network (RNN) is powerful to process sequential data in many fields including NLP [15] and speech recognition [9]. In details, RNN can change monotonously along with the position in a sequence. A rolled RNN structure can be unfolded as Fig. 3 illustrated. The hidden state $\boldsymbol{h}_t$ and output state $\boldsymbol{o}_t$ are updated as Eq. 2. In order to capture the relationship between trajectory points, we utilize a stacked RNN to learn the trajectory embeddings. And the experiments of visualization of trajectory embeddings illustrate the effectiveness of embedding.

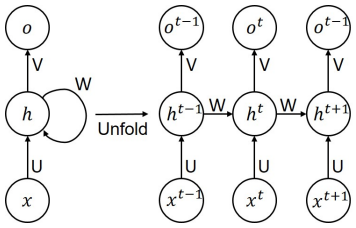


Fig. 3. Recurrent neural network.

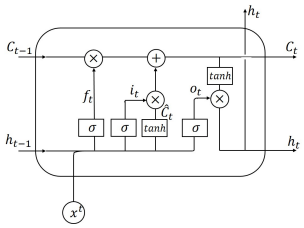


Fig. 4. The structure of LSTM.

$$\begin{aligned} \boldsymbol{h}_t &= f(\boldsymbol{W} \cdot \boldsymbol{h}_{t-1} + \boldsymbol{U} \cdot \boldsymbol{x}_t + \boldsymbol{b}_h), \\ \boldsymbol{o}_t &= f(\boldsymbol{V} \cdot \boldsymbol{h}_t + \boldsymbol{b}_o), \end{aligned} \tag{2}$$

where $\boldsymbol{W}$, $\boldsymbol{U}$, $\boldsymbol{V}$ are weight matrices, and $\boldsymbol{b}_h, \boldsymbol{b}_o$ are biases for hidden state and output state respectively. f is the activation function, and $\boldsymbol{x}_t$ is the embedding for each trajectory points.

Long short-term memory (LSTM) is a special kind of RNN cell, capable of learning long-term dependencies. The structure of LSTM cell is shown in Fig. 4. Specifically, the LSTM cell utilizes three gates to control the cell state. The forget gate $\boldsymbol{f}_t$ decides how much information of the last cell state $\boldsymbol{C}_{t-1}$ will keep

to the current state $\boldsymbol{C}_t$. The input gate $\boldsymbol{i}_t$ and $\hat{\boldsymbol{C}}_t$ decides how much information of the input $\boldsymbol{x}_t$ will keep to the current state $\boldsymbol{C}_t$. And the output gate $\boldsymbol{o}_t$ decides how much information of the current state $\boldsymbol{C}_t$ will output to $\boldsymbol{h}_t$. The detail information can be seen in Eq. 3,

$$\begin{aligned}
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f), \\
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i), \\
\hat{\boldsymbol{C}}_t &= tanh(\boldsymbol{W}_C \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_C), \\
\boldsymbol{C}_t &= \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \hat{\boldsymbol{C}}_t, \\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o), \\
\boldsymbol{h}_t &= \boldsymbol{o}_t * tanh(\boldsymbol{C}_t),
\end{aligned} \tag{3}$$

where $\boldsymbol{W}_f,\ \boldsymbol{W}_i,\ \boldsymbol{W}_C,\ \boldsymbol{W}_o$ are the weight matrices and $\boldsymbol{b}_f,\ \boldsymbol{b}_i,\ \boldsymbol{b}_C,\ \boldsymbol{b}_o$ are biases of each gate, respectively. σ and $tanh$ refer to the logistic sigmoid and hyperbolic tangent function.

Gated recurrent unit (GRU) is a gating mechanism in recurrent neural network which is similar to LSTM. A GRU has two gate, reset gate and update gate. Formally, the reset gate and update gate are calculated as Eq. 4,

$$\begin{aligned}
\boldsymbol{z}_t &= \sigma(\boldsymbol{W}_z \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\
\boldsymbol{r}_t &= \sigma(\boldsymbol{W}_r \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\
\tilde{\boldsymbol{h}}_t &= tanh(\boldsymbol{W} \cdot [\boldsymbol{r}_t * \boldsymbol{h}_{t-1}, \boldsymbol{x}_t]), \\
\boldsymbol{h}_t &= (1 - \boldsymbol{z}_t)\boldsymbol{h}_{t-1} + \boldsymbol{z}_t \tilde{\boldsymbol{h}}_t,
\end{aligned} \tag{4}$$

where $\boldsymbol{W}_z,\ \boldsymbol{W}_r,\ \boldsymbol{W}$ are weight matrices and $\tilde{\boldsymbol{h}}_t$ is the candidate state.

Learn the Trajectory Embedding In our proposed method, we utilize a delicately designed stacked RNN to learn trajectory embeddings, as shown in Fig. 2. The mapped trajectories are inputed into the stacked RNN sequentially. In every step, the RNN cell can learn a hidden state which summarizes the past sequences by merging the current step input information and the previous hidden state. The stacked RNN not only learns sequential information through the time step, but also memorizes significant information through the RNN layers which use non-linear function to capture trajectory characteristics in high-dimension space. Moreover, to avoid the over-fitting problem, dropout techniques [6] are used to constrain the representation ability of this model. In details, when data information flow pass through different RNN cells, we randomly select some edge and cut off the connection between the stacked layers. After the last step, we can get the trajectory embedding by concatenating the output state of the stacked RNN cells as Eq. 5:

$$\boldsymbol{E}_i = [\boldsymbol{h}_t^1,\ \boldsymbol{h}_t^2,\ \cdots,\ \boldsymbol{h}_t^l], \tag{5}$$

where $\boldsymbol{E}_i$ is the embedding for tr_i, l is the number of stacked RNN layers, and $\boldsymbol{h}_t^k$ is the last output state at layer k ($k = 1..l$).

4.4 Anomalous Detection

In section 4.3, we obtain the embedding $\boldsymbol{E}_i$ for the i-th trajectory. In order to detect anomalous, we utilize a multilayer perceptron (MLP) to reduce the dimensions of trajectory embedding.,

$$\boldsymbol{M}_i = \sigma(\boldsymbol{W}_i \cdot \boldsymbol{E}_i + \boldsymbol{b}_i) \tag{6}$$

where $\boldsymbol{M}_i$ is output vector, and $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$ are the projection parameters of the MLP layer. After that, as illustrated in Fig. 2, the results are fed into a softmax layer to generate the anomalous probability $\hat{y_i}$ of a trajectory, then given a trajectory datasets $D = \{tr_1, tr_2, \cdots, tr_m\}$, we can train our model by minimize the cross-entropy as following:

$$J = -\sum_{i=1}^{m}[y_i \ log \ \hat{y_i} + (1 - y_i) \ log(1 - \hat{y_i})]. \tag{7}$$

5 Experiments

5.1 Dataset and Metrics

We conduct all experiments on a real-world taxi trajectory dataset which is collected by 442 taxi in the city of Porto[1], in Portugal from Jan. 07, 2013 to Jun. 30, 2014 and the average sampling rate is $15s/points$. We extract 5 SD-pairs with sufficient historical trajectories. The basic information of those SD-pairs is shown in Table 1. There are about 5% anomalous trajectories.

Table 1. The information of SD-pairs.

	#Trajectories	#Anomalousness(%)	#avgPointsNum
SD-Pair1	1233	54(4.4%)	32
SD-Pair2	765	28(3.7%)	31
SD-Pair3	617	37(6.0%)	61
SD-Pair4	1379	44(3.2%)	51
SD-Pair5	4973	270(5.4%)	67

Instead of labelling the anomalous manually, we follow the solution in [20] which adopts a complete-linkage clustering algorithm to hierarchically cluster the trajectories. Given the prediction and its ground truth, we compute the measure metrics as Eq. 8,

[1] https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i/data

$$\begin{aligned} ACC &= \frac{TP+TN}{TP+TN+FP+FN}, \\ P &= \frac{TP}{TP+FP}, \\ R &= \frac{TP}{TP+FN}, \\ F_1 &= \frac{2PR}{P+R}, \end{aligned} \tag{8}$$

where TP, TN, FP and FN are true positive, true negative, false positive and false negative in confusion matrix, respectively.

5.2 Baselines

We compare our method with the baselines based on density and isolation. The baselines are as following:

1. LCS [21]: The Longest Common Sub-sequence (LCS) method matches the longest common sub-sequence between two sequences using dynamic programming. LCS is also a widely used representative method for measuring the trajectory similarity. We implement LCS by comparing all trajectories in the training set for every given testing trajectory.

2. XGBoost [5]: XGBoost is an ensemble model based on the gradient boosting decision tree (GBDT) and is designed to be highly efficient, flexible, and portable. Those characteristics we extracted from a trajectory contain the distance of a trajectory, the angle between trajectory points and so on.

3. TOP-EYE [8]: TOP-EYE uses a decay function to mitigate the influences of the past trajectories on the evolving outlying score, which is defined based on the evolving moving direction and density of the historical trajectories. Since the sampling rate in our dataset is $15s/points$, we conduct this method by counting the density of each grid and compute anomalous score for each test trajectory.

4. IBOAT [4]: Anomalous trajectories will be isolated from the majority of routes, while normal trajectories will be supported by a large number of trajectories. IBOAT adopts the inverted index mechanism to fast retrieve the relevant trajectories.

5.3 Settings

We implement the proposed model ATD-RNN with TensorFlow [1] and run the code on an Intel Core i7-4790 with 8-GB RAM. As we introduced in section 4, there are two kinds of RNN cell. In our experiments, ATD-LSTM represents the model using the LSTM cell and ATD-GRU represents the model using GRU cell. There are three key parameters in our model, the embedding dimensions, the number of RNN layers and the dropout probabilities. For generality, we set the dimensions of each point to 64, the size of hidden state of a RNN cell to 64 and the layers of stacked RNN to 5. To alleviate the problem of over-fitting inherent of RNN, we apply dropout technique [6] and the dropout probability is set to 0.5. We utilize Adam [10] to optimize our model.

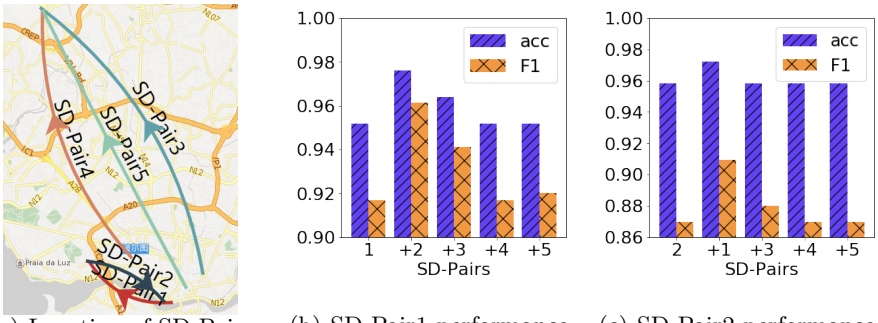
5.4 Results and Analysis

Performance Evaluation We evaluate the results on SD-Pair with the above baselines. The results of anomalous trajectory detection in given SD-pair are shown in Table. 2. We can observe that in most cases, ATD-LSTM and ATD-GRU achieve the best results. This indicates that recurrent neural network could capture the internal characteristics of anomalous trajectories and normal trajectories. The better performance of ATD-RNN shows that the sequential information of a trajectory is critical to anomalous trajectory detection. In addition, the reason of the undesirable results of LCS and XGBoost may be that those methods only consider the shape of a trajectory and neglect the information of the historical dataset and the sequential information of the trajectory. On the contrary, TOP-EYE and iBOAT achieve considerable performance because those methods utilize the historical trajectories. And the reason of iBOAT outperforms TOP-EYE in most circumstances. The reason is probably that iBOAT not only takes the similarity of historical trajectories into consideration, but also makes use of the local sequential information in the trajectory.

Table 2. Performance evaluation of anomalous trajectory detection on different SD-Pairs. ATD-LSTM is ATD-RNN based on LSTM cell, and ATD-GRU is ATD-RNN based on GRU cell.

		LCS	XGBoost	TOP_EYE	iBOAT	ATD-LSTM	ATD-GRU
SD-Pair1	Acc	0.8434	0.8795	0.9629	0.9506	0.9518	**0.9638**
	P	0.6765	0.9412	0.9583	0.9565	0.9565	0.9231
	R	0.9200	0.6400	0.9200	0.8800	0.8800	0.9600
	F_1	0.7797	0.7619	0.9230	0.9167	0.9167	**0.9412**
SD-Pair2	Acc	0.9444	0.8810	0.9444	**0.9583**	**0.9583**	**0.9583**
	P	0.9000	1.0000	0.7500	0.9091	0.9091	0.8462
	R	0.7500	0.2857	1.0000	0.8333	0.8333	0.9167
	F_1	0.8182	0.4444	0.8571	0.8696	0.8696	**0.8800**
SD-Pair3	Acc	0.9625	0.8375	0.9625	0.9625	**0.9875**	0.9750
	P	0.9333	0.7143	0.9333	0.8824	0.9412	0.8889
	R	0.8750	0.3125	0.8750	0.9375	1.0000	1.0000
	F_1	0.9032	0.4348	0.9032	0.8276	**0.9697**	0.9412
SD-Pair4	Acc	0.9242	0.9470	0.9470	0.9394	0.9848	**0.9924**
	P	1.0000	0.9412	0.8261	1.0000	0.9545	1.0000
	R	0.5455	0.7273	0.8636	0.6364	0.9545	0.9545
	F_1	0.7059	0.8205	0.8444	0.7778	0.9545	**0.9767**
SD-Pair5	Acc	0.7819	0.7574	0.8431	0.8750	0.9020	**0.9167**
	P	0.9623	0.9756	0.6832	0.8000	0.9537	0.9063
	R	0.3696	0.2898	1.0000	0.8406	0.7463	0.8406
	F_1	0.5340	0.4469	0.8118	0.8198	0.8374	**0.8722**

Study on Influences with Multi-SD-Pairs. To explore the influence of anomalous trajectory detection on different SD-Pairs instead of a given SD-pair,

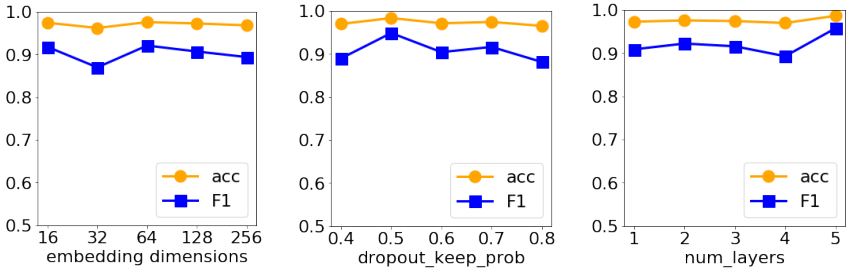


(a) Location of SD-Pairs. (b) SD-Pair1 performance. (c) SD-Pair2 performance.

Fig. 5. Anomalous trajectory detection with Multi-SD-pairs. "1" means the model on SD-Pair1, and "+1" means the model of some SD-Pair and SD-Pair1.

we conduct experiments on the union set of two SD-Pairs selected from SD-pair1 to SD-pair5. In details, for SD-Pair1 and SD-Pair2, we add the remaining SD-Pairs into the train process, and then test the performance on SD-Pair1 and SD-Pair2 respectively. For example, we train a model with SD-Pair1 and SD-Pair3, and then test the performance on SD-Pair1. The relative location of these SD-Pairs is shown in Fig. 5(a). We can see that the source and destination of SD-Pair1 are close to SD-Pair2, while SD-Pair3, SD-Pair4 and SD-Pair5 are close to each other. The results are shown in Fig. 5. At the first sight, we can see that if we train a model with different SD-Pairs trajectories dataset, the results are better than a single SD-Pair. This demonstrates that the extra trajectory information is useful in anomalous trajectory detection. Concretely, the merged set of SD-Pair1 and SD-Pair2 gets a significant performance which indicates that the close SD-Pairs could enhance the results of anomalous trajectory detection. The probable reason is that different SD-Pairs which are close to each other can offer valuable information in detection process and trajectory from different SD-Pairs can deliver information through the model. As mentioned in Fig. 1(b), the trajectories from S_2 to D can provide information to the detection process from S_1 to D. In addition, the results indicate that the proposed model ATD-RNN has potential to alleviate data sparsity.

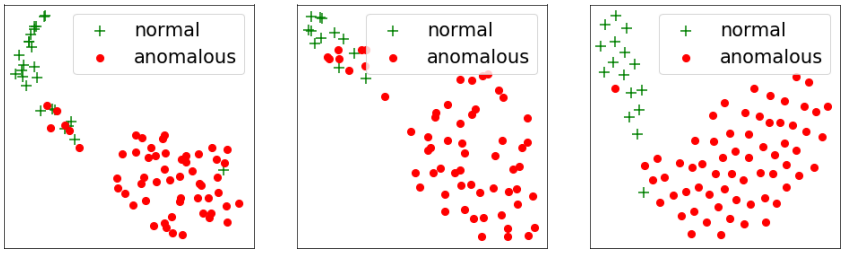
Parameter Experiments. In this section, we explore the influence of different parameters. The key parameters include the dimensions of embedding, the dropout probability and the number of stacked RNN layers. All parameter experiments are conducted on SD-Pair4 with LSTM cell. The results are shown in Fig 6. As shown in Fig 6(a), we can observe that the accuracy and F_1 get the best score when *embedding dimensions* is 64. As one can see from Fig 6(b), when *dropout_keep_prob* is 0.5, the model gets the best performance. It is interesting that the change of *dropout_keep_prob* has slightly influence on accuracy, but the influence on F_1 is serious. The reason of this phenomenon may cause by



(a) Embedding dimension. (b) Dropout probabilistic. (c) The number of layers.

Fig. 6. Performance of different parameters.

the unbalance of anomalous and normal trajectory proportion. As illustrated in Fig 6(c), when *num_layers* is 5, we can get a significant performance.



(a) SD-Pair1 (b) SD-Pair2 (c) SD-Pair3

Fig. 7. Visualization of trajectory embeddings.

Visualization. To analyze the trajectory embedding, we visualize the vectors of trajectory embedding by t-SNE [16]. The results are shown in Fig. 7. We can observe that the trajectories embeddings can separate the anomalous trajectories from normal historical trajectories dataset. The results indicate that the trajectory embedding learned by recurrent neural network can capture the internal characteristics of the trajectories.

6 Conclusions and Future Work

In this paper, we propose **A**nomalous **T**rajectory **D**etection using **R**ecurrent **N**eural **N**etwork (**ATD-RNN**). ATD-RNN learns the trajectory embeddings through a delicately designed recurrent neural network. Comparing to traditional

methods, ATD-RNN can capture the sequential information of the historical trajectories and the internal characteristics between the anomalous and normal trajectories. Moreover, ATD-RNN is not constrained by the given SD-Pair, but takes different SD-Pairs into consideration. This means that ATD-RNN has potential to alleviate data sparsity. Extensive experiments on real-world datasets demonstrate the effectiveness of ATD-RNN.

In the future, we will extend our model to learn the trajectory embedding with the up-to-date techniques, e.g., attention mechanism and memory augmentation. In addition, we will attempt to solve other trajectory data mining tasks and make use of the additional information to improve the quality of the trajectory embedding.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China (No. 61772082, 61702296, 61375058), and the Beijing Municipal Natural Science Foundation (4182043).

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: a system for large-scale machine learning. In: OSDI. vol. 16, pp. 265–283 (2016)
2. Bu, Y., Chen, L., Fu, A.W.C., Liu, D.: Efficient anomaly monitoring over moving object trajectory streams. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 159–168. ACM (2009)
3. Cai, Y., Wang, H., Chen, X., Jiang, H.: Trajectory-based anomalous behaviour detection for intelligent traffic surveillance. IET intelligent transport systems 9(8), 810–816 (2015)
4. Chen, C., Zhang, D., Castro, P.S., Li, N., Sun, L., Li, S., Wang, Z.: iboat: Isolation-based online anomalous trajectory detection. IEEE Transactions on Intelligent Transportation Systems 14(2), 806–818 (2013)
5. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
6. Gal, Y., Ghahramani, Z.: A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems. pp. 1019–1027 (2016)
7. Gao, Q., Zhou, F., Zhang, K., Trajcevski, G., Luo, X., Zhang, F.: Identifying human mobility via trajectory embeddings. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1689–1695. AAAI Press (2017)
8. Ge, Y., Xiong, H., Zhou, Z.h., Ozdemir, H., Yu, J., Lee, K.C.: Top-eye: Top-k evolving trajectory outlier detection. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 1733–1736. ACM (2010)
9. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. pp. 6645–6649. IEEE (2013)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

11. Knorr, E.M., Ng, R.T., Tucakov, V.: Distance-based outliers: algorithms and applications. The VLDB JournalThe International Journal on Very Large Data Bases 8(3-4), 237–253 (2000)
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: International Conference on Machine Learning. pp. 1188–1196 (2014)
13. Lee, J.G., Han, J., Li, X.: Trajectory outlier detection: A partition-and-detect framework. In: Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on. pp. 140–149. IEEE (2008)
14. Li, X., Han, J., Kim, S., Gonzalez, H.: Roam: Rule-and motif-based anomaly detection in massive moving object data sets. In: Proceedings of the 2007 SIAM International Conference on Data Mining. pp. 273–284. SIAM (2007)
15. Luong, M.T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025 (2015)
16. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research 9(Nov), 2579–2605 (2008)
17. Mnih, A., Kavukcuoglu, K.: Learning word embeddings efficiently with noise-contrastive estimation. In: Advances in neural information processing systems. pp. 2265–2273 (2013)
18. Sillito, R.R., Fisher, R.B.: Semi-supervised learning for anomalous trajectory detection. In: BMVC. vol. 1, pp. 035–1 (2008)
19. Wu, F., Fu, K., Wang, Y., Xiao, Z., Fu, X.: A spatial-temporal-semantic neural network algorithm for location prediction on moving objects. Algorithms 10(2), 37 (2017)
20. Wu, H., Sun, W., Zheng, B.: A fast trajectory outlier detection approach via driving behavior modeling. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 837–846. ACM (2017)
21. Ying, J.J.C., Lee, W.C., Weng, T.C., Tseng, V.S.: Semantic trajectory mining for location prediction. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. pp. 34–43. ACM (2011)
22. Yu, Y., Cao, L., Rundensteiner, E.A., Wang, Q.: Detecting moving object outliers in massive-scale trajectory streams. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 422–431. ACM (2014)
23. Zhang, D., Li, N., Zhou, Z.H., Chen, C., Sun, L., Li, S.: ibat: detecting anomalous taxi trajectories from gps traces. In: Proceedings of the 13th international conference on Ubiquitous computing. pp. 99–108. ACM (2011)
24. Zhao, W.X., Zhou, N., Sun, A., Wen, J.R., Han, J., Chang, E.Y.: A time-aware trajectory embedding model for next-location recommendation. Knowledge and Information Systems pp. 1–21 (2017)
25. Zheng, Y.: Trajectory data mining: an overview. ACM Transactions on Intelligent Systems and Technology (TIST) 6(3), 29 (2015)
26. Zhu, J., Jiang, W., Liu, A., Liu, G., Zhao, L.: Time-dependent popular routes based trajectory outlier detection. In: International Conference on Web Information Systems Engineering. pp. 16–30. Springer (2015)