Hindawi
Security and Communication Networks
Volume 2019, Article ID 8431074, 12 pages
https://doi.org/10.1155/2019/8431074

Research Article

Detection of Dummy Trajectories Using Convolutional Neural Networks

Jiaji Pan,[1] Yining Liu,[1] and Weiming Zhang[2]

[1]*School of Computer and Information Security, Guilin University of Electronic Technology, China*
[2]*CAS Key Laboratory of Electromagnetic Space Information, University of Science and Technology of China, Hefei 230026, China*

Correspondence should be addressed to Yining Liu; lyn7311@sina.com

Received 25 January 2019; Revised 13 March 2019; Accepted 17 April 2019; Published 2 May 2019

Guest Editor: Fagen Li

Nowadays, privacy in trajectory is an important issue in the coming big data era. In order to provide better protection for trajectory privacy, a number of solutions have been proposed in the literature, and the dummy trajectory method has attracted great interests in both academia and industry recently due to the following advantages: (1) neither a third-party server nor other parties' cooperation is necessary; (2) location-based services are not influenced; and (3) its algorithm is relatively simple and efficient. However, most of trajectory privacy generations usually consider the geometric shape of the trajectory; meanwhile the real human mobility feature is usually neglected. In fact, the real trajectory is not the product of random probability. In this paper, convolutional neural network (CNN) is used as the learning machine to train with lots of the real trajectory and the generated dummy trajectory sets. Then, the trained classifier is used to distinguish the dummy from the real trajectory. Experiments demonstrate that the method using CNN is very efficient, and more than 90% of dummy trajectories can be detected. Moreover, the real trajectory erroneous judgment rate is below 10% for most of real trajectories.

1. Introduction

Nowadays, location-based services (LBS) [1–7] are widely used in smart mobile terminals, which makes the peoples' daily life more convenient. In the process of interaction between mobile terminal and LBS, there are many schemes to protect users' personal privacy data [8–10]. In addition, the large amount of trajectory data generated in the interaction process consists of the abundant space-time information such as users' personal interests, economic status, and living habits. In fact, this information is sensitive, and it should not be directly released to the public. On the other hand, the trajectory data is useful for the municipal transportation service, decision-making of government, and other business applications. Therefore, how to protect the data privacy [11, 12], especially to balance single trajectory privacy and the trajectory data publishing, is an interesting research topic, and it has attracted attentions all over the world.

The existing trajectory privacy protection methods include dummy trajectory method [13–16], trajectory suppression method [17], generalization method [6, 18], and differential method [19, 20].

For dummy trajectory method, several dummy trajectories are generated for each real trajectory; then the real trajectory and $k-1$ dummy trajectories are published together to reduce the probability of true trajectory exposure.

For trajectory suppression method, the sensitive information in the real trajectory is not released in order to protect the user's personal information contained in the trajectory, which needs to set or process the sensitive information in advance. Therefore, how to find the sensitive information becomes the key issue of the suppression method. Intuitively speaking, when there is a clear need to suppress the information, suppression method is simple and effective and can achieve the purpose of user privacy protection by simple data processing [17]. However, this method relies on the determination of sensitive information, that is, how to suppress the sensitive needs to know what resource is owned by the opponent. Obviously, it is not easy. In addition, the simple and crude direct deletion of sensitive information will reduce the usability of trajectory data.

For generalization method, its basic idea is to generalize the QI (Quasi identifier) attributes that can uniquely identify

the user, which guarantees that the real trajectory cannot be distinguished from other trajectories. The k-anonymity model is commonly used in trajectory privacy, which converts the D (trajectory data) in the database into D∗, so that any trajectory T in D∗ belongs to a trajectory k-anonymity set, and the information distortion between D∗ and D is minimized. In LBS, how to choose the scope of anonymous boxes is not easy, since the real trajectory is not known in advance.

For the differential method, it is used in trajectory privacy protection in recent years. It does not need to consider the background knowledge of the opponent and it is based on strict mathematical knowledge. It provides a quantifiable, assessable, and provable method for privacy protection. By adding random noise perturbation sensitive data, it can distort some data while maintaining its statistical properties.

In recent years, the dummy trajectory method is widely researched due to the following reasons:

(1) No third-party server is required, which makes it more robust.

(2) The algorithm of dummy trajectories generation is relatively simple and efficient.

(3) The service based on the precise location is not influenced since the real trajectory is kept.

There are many algorithms for generating dummy trajectory, such as rotation method, intraregion random point method, translation method, and the combination of these methods. The background factors are also taken into account in the generation of dummy trajectories. The basic idea of dummy trajectory was first proposed by Kido et al. [18, 21], in which there are two dummy trajectory design rules and the generation method. Lei et al. [14] proposed a method to increase the number of dummy trajectories by adding intersections to the trajectories obtained after rotation, therefore improving the privacy protection level of the real trajectory. In Wu et al.'s scheme [15], not only the distance between the real trajectory and the dummy trajectory is involved in the dummy trajectory generation, but also the distance between the dummy trajectories is also considered. By disturbing the generated dummy trajectories, the final set of trajectories can satisfy the privacy requirements. Kato R et al. [16] assumed that the user's movements are known in advance and proposed a dummy-based anonymization method based on the predicted movement, where dummies move naturally while stopping at several locations. Niu et al. proposed a dummy location generation algorithm based on background information, especially the probability of sending requests in each location being considered, and ensure that the generated $(k - 1)$ dummy locations can more easily confuse the opponent by formalizing the background information [22]. Hara et al. have further studied the problem of trajectory privacy protection in mobile vehicle network and designed a dummy trajectory generation algorithm based on vehicle trajectory [23]. Since the algorithm takes into account the trajectory characteristics of vehicle movement, the probability of dummy trajectory being guessed is reduced. Lu et al. [24] pointed out that although the generated dummy has a high density distribution, it can reduce the protection degree of users' location privacy. They divide the circle/grid region into subregions of equal size and distribute all positions over different radii/vertices. In literatures [25, 26], the authors focus on the real environment requirements considering physical constraints and propose a new virtual generation algorithm DumGrid and Dum-P [25]. Dum-P generates dummies around the user in grid mode; it ensures that the more realistic movement model generates dummies with the user's movement. To solve the problem of insufficient location privacy requirements, an improved version Dum-P-Cycle is proposed in [26]. In addition, perhaps chaotic system is a feasible tool to protect the trajectory information [27].

However, the human mobility model is not involved in most of the current dummy trajectory generation algorithms. In reality, the trajectory is not a set of random points but a set of points satisfying some known or unknown features, which is constrained by various conditions, especially people's behavior factors. For example, the deviation angle between the trajectory segment and the segment is usually very small (people tend to go straight); when the deviation angle suddenly increases (such as car turning), it means that the length of the following segments begins to decrease (such as the speed of driving turns decreased) and so on. In short, the trajectory points are arranged according to some certain rules, and these rules are often restrictive. Specifically, the real trajectory is usually purposeful. For the sake of efficiency, the trajectory segment consisting of the set of trajectory points generally has a small deflection angle and few frequent oscillations. People usually move with a uniform speed, so the length of trajectory segment should be gentle. However, most of the dummy trajectories generation relies on the use of a random method, the dummy trajectory deviation is frequent, and the length of the trajectory segment is oscillatory; therefore the trajectory points often fall in the nonreachable place. In conclusion, the current dummy trajectories generation algorithms do not take into account the behavioral characteristics contained in real trajectories; therefore there is a considerable probability of identifying dummy trajectories by analyzing the distribution characteristics of real trajectories.

In addition, there are complex laws between the points of real trajectories and dummy trajectories. The difference between real trajectory and dummy trajectory is difficult to be represented by simple function. Artificial neural network is a feasible tool, since it can be used as classifier in large-scale data training. Moreover, this process does not require much people interaction, and the parameters are automatically generated through a lot of iterative learning. As long as hyperparameters and network models are set reasonably, good classification results can be achieved.

In this paper, CNN model is used to generate the classification function, and the overall framework of our dummy trajectory detection method is shown in Figure 1.

We define a series of trajectory points to form a trajectory section, and two adjacent points form a trajectory segment. When a trajectory is detected, it is divided into sections of equal trajectory points at first. If the last section is short of the point number of trajectory section, the rounding method is adopted. In other words, less than half of the point number

$seg_i =< p_i, p_{i+1} > (i = 1, 2...k-1)$

$sec_i = \{seg_j, seg_{j+1}, ...seg_{j+m-1}\}\ \ (j = (i-1)\,m + 1)\ \left(i = 1, 2, ... \left\lfloor \frac{k}{m} \right\rfloor\right)$

If $(k\%m)/m >= 0.5$**:**

$sec_{\lfloor k/m \rfloor + 1} = \{seg_{k-m+1}, seg_{k-m+2}, ...seg_k\}$

Else: Round the rest segments

Algorithm 1

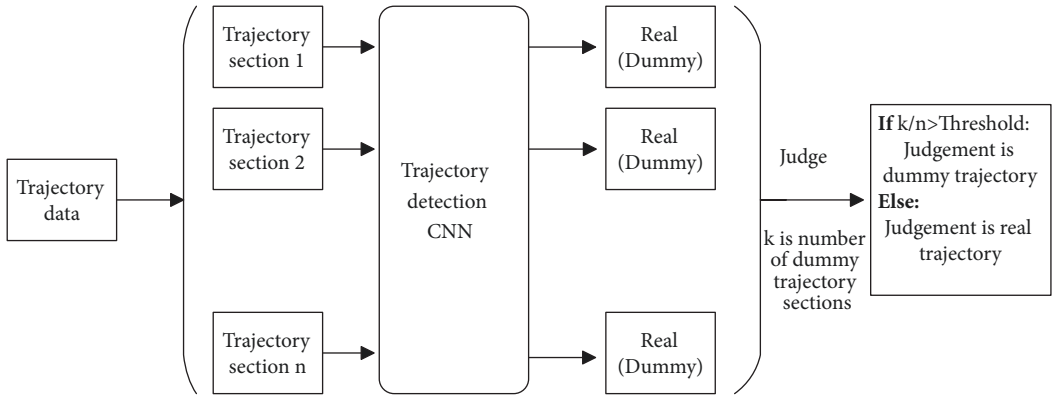


Figure 1: Dummy trajectory detection model framework based on CNN.

of trajectory section is rounded. Otherwise, the point is taken from the back to the front to reach the point number of trajectory section. For simplicity, in this paper, we suppose that a trajectory consists of m segments, and each m segment forms a section. The above concepts are defined as shown in Algorithm 1.

As shown (in which figure or algorithm), we obtain n trajectory sections; then we put n trajectory sections into the trained CNN, respectively, and obtain the judgment result of n trajectory sections. Suppose that k sections are judged as dummy; the trajectory is judged as the dummy when k/n is more than the predefined threshold. The lower the threshold is, the stricter the dummy trajectory detection is and the higher the dummy trajectory detection rate is but the higher the real trajectory erroneous judgment rate is.

The main contributions of our work are summarized as follows:

(1) Unlike previous works that set privacy standards and trajectory parameters to generate dummy trajectories, we try to find the differences between real trajectories and dummy trajectories from the attacker's point of view, which is useful for improving the dummy trajectory generation.

(2) The deep learning is used to train the behavior feature classifier of the human's movement; then the classifier is used to distinguish the dummy trajectories that are generated according to the current main algorithms.

2. Preliminaries

2.1. Trajectory Representation Method

Absolute Trajectory. The absolute trajectory consists of a series of trajectory points with latitude and longitude as the spatial metric, with time series as the trajectory points arrangement order, which is defined as $traj_{abs} = \{< loc_1(\mathrm{lat}_1,\mathrm{lng}_1),\mathrm{t}_1 >, < loc_2(\mathrm{lat}_2,\mathrm{lng}_2),\mathrm{t}_2 > ... < loc_n(\mathrm{lat}_n,\mathrm{lng}_n),\mathrm{t}_n >\}$. The trajectory data set released by major institutions is also an absolute trajectory.

Relative Trajectory. Although the absolute trajectory can accurately express the position of the trajectory on the earth's surface, it is not convenient to manipulate the trajectory such as stretching and rotating and to calculate some trajectory characteristics. Longitude and latitude can be regarded as absolute coordinates. The relative coordinate system describing the relative trajectory is a Cartesian coordinate system in which the point specified on the plane of the map where the trajectory is located is the coordinate zero and the direction specified is the x-axis and y-axis. In the process of trajectory data processing, we need to transform absolute trajectory into relative trajectory. The relative trajectory is defined as follows:

$$traj_{rel} = \{< loc_1(x_1,\mathrm{y}_1),\mathrm{t}_1 >, < loc_2(x_2,\mathrm{y}_2),\mathrm{t}_2 > ... < loc_n(x_n,\mathrm{y}_n),\mathrm{t}_n >\}$$

Feature Trajectory. Unlike the above two trajectory definitions, which use trajectory points to define trajectories, the feature trajectory is defined using trajectory shape features including the relative offset angles (roa) and relative lengths (rl) as follows:

$$traj_{fea} = \{< roa_1 = 0, rl_1, \mathrm{t}_1 >, < roa_2, rl_2, \mathrm{t}_2 > ... < roa_n, rl_n, \mathrm{t}_n >\}$$

2.2. Convolutional Neural Network. CNN is a kind of multilayer neural network, which is good at dealing with machine

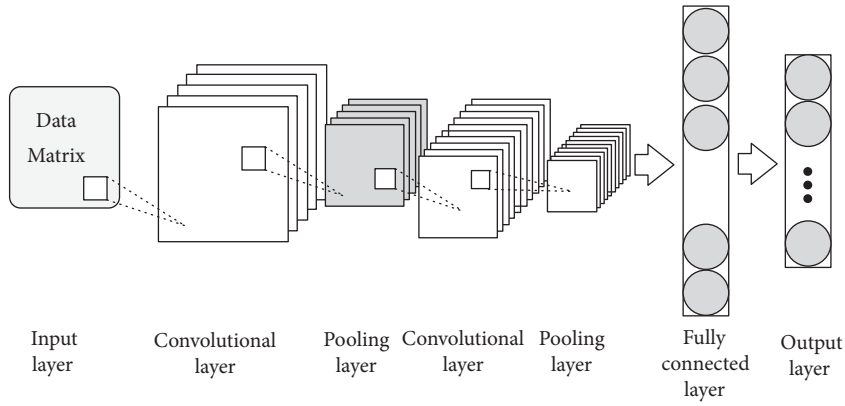


Figure 2: Classic CNN structure diagram.

learning problems related to images, especially large images [28]. CNN reduces the dimension of image recognition problem by a series of methods and makes it possible to be trained eventually. CNN consists of input layer, convolutional layer, activation function, pooling layer, and fully connected layer [29]. Inspired by the concept of local receptive field (participating in a convolutional kernel operation is an area of the input image (or feature map), the size of which is the receptive field), the convolutional layer connects the input small area and calculates the dot multiplication between the convolutional core and the corresponding input small area as the output. While the pooling layer is mainly designed to reduce the data dimension, it performs a downsampling operation on the spatial dimension (width, height). The classic CNN structure diagram is shown in Figure 2.

The convolutional layers are used for feature extraction, and we traverse the input matrix with a matrix called filter. The number of output matrices after each convolutional operation is the number of filters. The pooling layer compresses the input feature map. On one hand, it reduces the feature map and simplifies the network computing complexity; on the other hand, it compresses the feature and extracts the main features (we usually use maximum pooling). Fully connected layer connects all features and sends the output value to the classifier (such as Softmax classifier).

In summary, CNN extracts the features through convolutional layer and reduces the parameters and computational times through pooling layer. In fact, it completes classification tasks by traditional neural network. Compared with other classifiers, its filter used to extract data features has the characteristic of weight sharing. The adjacent trajectory feature metrics we need to extract have similar characteristics and we can share weights to extract. Therefore, CNN is compatible with our classification tasks.

3. Training and Detection

3.1. Feature Trajectory Definition

3.1.1. Relative Offset Angle. As shown in Figure 3, $\overrightarrow{a}$, $\overrightarrow{b}$, and $\overrightarrow{c}$ are three trajectory vectors and $\overrightarrow{d}$ is the extension line of vector $\overrightarrow{a}$. The vector angle θ_1 of vector $\overrightarrow{b}$ and vector $\overrightarrow{a}$ is defined as relative offset angle; also, the relative offset angle of $\overrightarrow{c}$ relative to $\overrightarrow{a}$ is θ_2. If $\overrightarrow{b}$ is over the vector $\overrightarrow{d}$, the offset angle is positive; otherwise it is negative.

3.1.2. Relative Length. We assume that the total length of the trajectory section is s, the i^{th} trajectory segment in the section is s_i, and the relative length of the i^{th} trajectory segment is defined as s_i/s.

3.2. Feature Trajectory Generation Algorithm. We use Algorithm 2 to generate the feature metrics of trajectory sections. No matter how the trajectory rotates or how the trajectory sections are scaled equally, the trajectory feature metric will not change. As long as the trajectory shape is the same, we regard it as the same trajectory. Similarly, for dummy trajectories, we also use this method to produce feature metrics of trajectory section. After generating the feature trajectories, we put the real and dummy trajectory set into the detector for training to initialize the detector.

3.3. Trajectory Data Preprocessing. In trajectory data preprocessing, the absolute trajectory is transformed to relative trajectory and then is transformed into feature trajectory. We take 5s as the time interval to extract a continuous series of points (longitude and latitude representation) from the testing trajectories. Then we use Algorithm 2 to transform relative trajectory (RT) into feature trajectory (FT). As shown in Table 1, we use RT and FT to represent a trajectory segment jointly.

3.4. Detector Design and Training Process

3.4.1. Detector Model Structure. There is a strong correlation between the trajectory segment and the trajectory segment; we use the correlation between the trajectory segments with CNN. Most of the other depth neural networks consider the data characteristics from the input data as a whole and cannot extract the trajectory characteristics very well. Compared with other artificial neural networks, the filter of CNN has unique weight sharing characteristics and we also need the sharing weights to extract feature of adjacent trajectory segments; therefore, CNN is used as classifier in

Input: k points intercepted on the trajectory (x_i, y_i), for i = 1 to k.
Output: k-2 relative offset angles (roa) and k-1 relative length (rl)

(1) $dis_{sum} = 0$
(2) **for** i= 1 to k-1 **do**
(3) $\overrightarrow{vec_i} = (x_{i+1} - x_i, y_{i+1} - y_i)$
(4) $dis_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$
(5) $dis_{sum} += dis_i$
(6) **end for**
(7) **for** j = 1 to k-2 **do**
(8) $roa_j = \arccos\left(\dfrac{\overrightarrow{vec_{j+1}}\overrightarrow{vec_j}}{\left|\overrightarrow{vec_{j+1}}\right|\left|\overrightarrow{vec_{j+1}}\right|}\right)$
(9) $k = \dfrac{y_{j+1} - y_j}{x_{j+1} - x_j}$
(10) $b = y_j - kx_j$
(11) **if** $y_{j+2} > kx_{j+2} + b$
continue
(12) **else**
$roa = -roa$
(13) **end for**
(14) **for** t = 0 to k-1**do**
(15) $rl_t = \dfrac{dis_t}{dis_{sum}}$
(16) **end for**
(17) **Return** $traj_{fea} = \{< 0, rl_1 >, < roa_1, rl_2 > \ldots < roa_{k-2}, rl_{k-1} >\}$

Algorithm 2: Trajectory feature metrics generation.

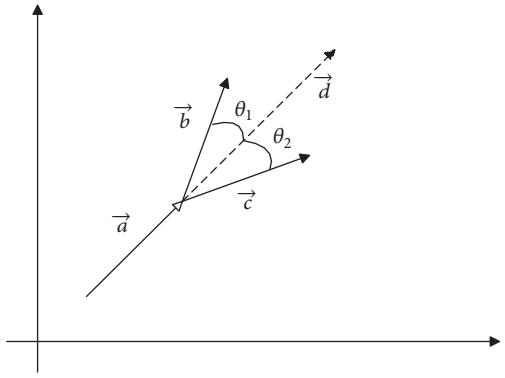


Figure 3: Relative offset angle.

Table 1: Representation of trajectory segment.

Trajectory section			
RT		FT	
x_1	y_1	roa_1	rl_1
x_2	y_2	roa_2	rl_2
$\ldots$	$\ldots$	$\ldots$	$\ldots$
x_{m-1}	y_{m-1}	roa_{m-1}	rl_{m-1}

this paper. Unlike ordinary image processing, we deal with the feature matrix here and need to make some improvements to the universal network model. The network model structure and training process are shown in Figure 4. The CNN model in this paper is the improvement of the universal model of CNN introduced in literature [30], especially in the network layer architecture of convolutional layer and pooling layer. The detailed steps in the dotted box of Figure 4 are depicted in Figure 5.

The process of doing one network training is as follows:

(1) Relative trajectory (RT) and feature trajectory (FT) are jointly used to represent a trajectory segment; they are input into convolutional layer of CNN to extract trajectory features.

(2) Maximum pooling operations are used to reduce the amount of parameters after each convolutional operation; then all outputs are connected to a matrix for each feature metric.

(3) Fully connected layer is used to synthesize the features extracted from the front to obtain a 1×2 matrix.

(4) Softmax and cross entropy operations (Softmax operation calculates the probability value of classification results; cross entropy calculates the distance between CNN classification results and real classification, which is called loss) are executed on the matrix obtained by the fully connected layer to get loss. The total Softmax is defined as Softmax = αSoftmax1 + $(1 - \alpha)$Softmax2, where α is the weight.

(5) Convolutional layer and fully connected layer parameters W and b (W is a weight parameter and b is a bias

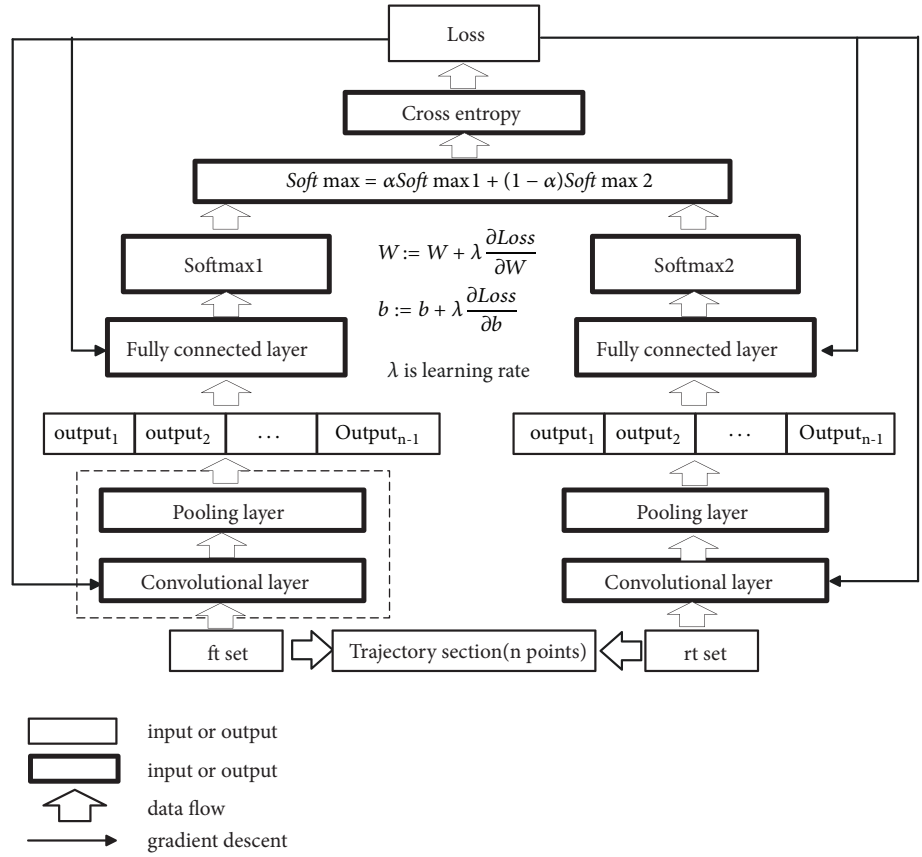


Figure 4: Training process for CNN.

item) are updated by gradient descent method based on loss equation.

FT and RT describe trajectories in different ways, and their dimensions are different, so they cannot be put together. The network structure on the left is FT feature detection network and the one on the right is RT feature detection network. For each iteration process, we send the corresponding RT and FT to networks separately. Then we use gradient descent method (Gradient descent is one of the iterative methods, which can be used to solve the least squares problem. The calculation of gradient descent method is to solve the minimum value along the direction of gradient descent.) to update the weight parameter W and bias parameter b of convolutional layer and fully connected layer. After updating the network model for a certain number of times, the model can distinguish the real trajectory from the dummy trajectories.

In these two layers, the following operations are executed:

(1) We arrange and transform feature trajectory into matrix in time sequence. Feature trajectory is represented as a $(n-1)\times 2$ matrix as illustrated in the right part of Table 1.

(2) We use n-1 types of filters to extract the arrangement characteristics of feature trajectory, and there are 128 filters in each type. For the first k type filter, we extract characteristics of adjacent k trajectory segments in turn by convolutional operation to obtain a $(n-k)\times 1$ matrix. Then $128n-128$ matrices that are output from convolutional operation are, respectively, executing the maximum pooling operation.

(3) After operation of a pooling layer, we obtain $n-1$ outputs and each output has 128 matrices. Then, we connect these $n-1$ outputs into a matrix and send them to the fully connected layer shown in Figure 4 for the next step.

3.4.2. Trajectory Detection Schemes. The network architecture of dummy trajectory detection is shown in Figure 6. Different from the CNN training process, the forward propagation process is only executed once to obtain Softmax; then the final judgment is made according to Softmax.

We first need to express the detected trajectory sections with FT and RT, respectively, and then put them into CNN to run in the direction indicated by the arrow once. Finally, we can obtain the detection result.

4. Experiments and Analysis

We use the trajectories from Microsoft research GeoLift project as the testing data set; 182 pieces of users' track data were collected from April 2007 to August 2012. These data

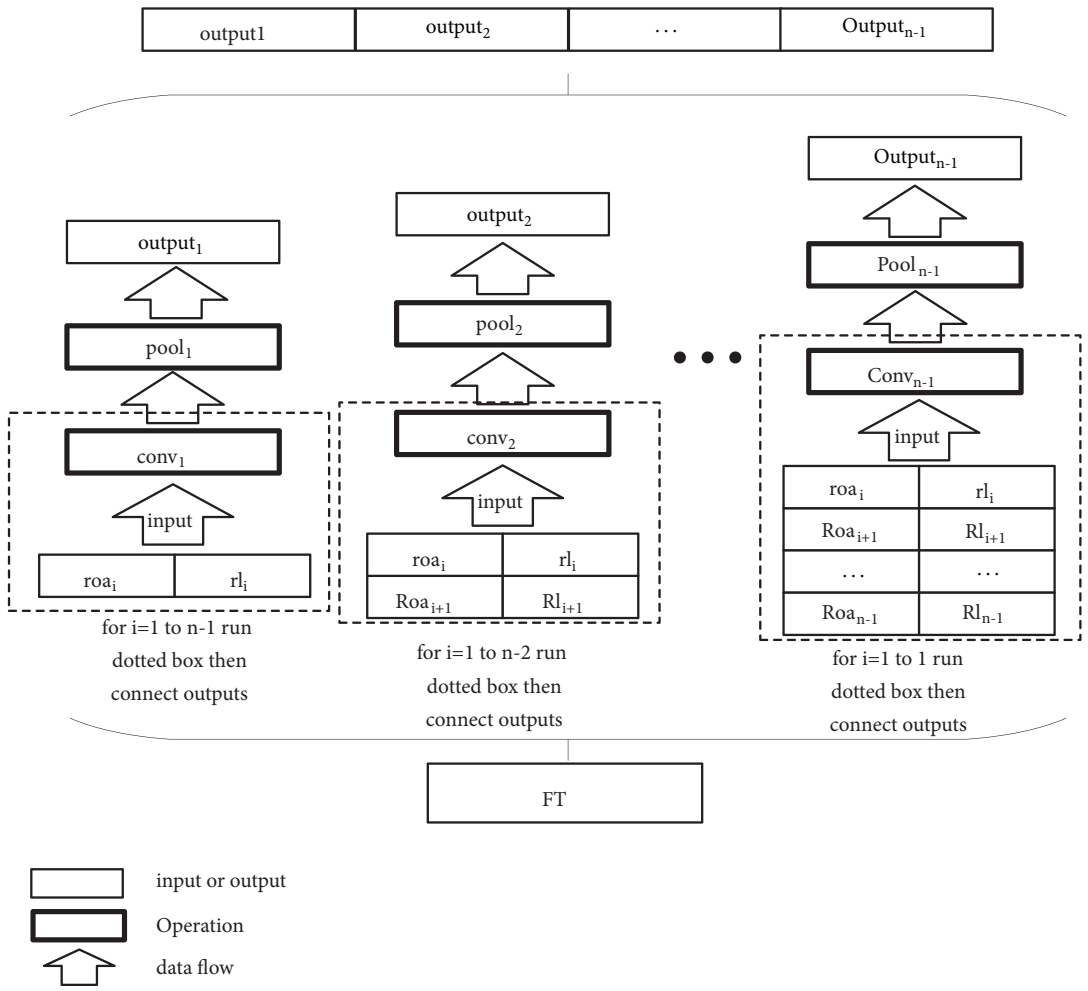


Figure 5: The convolutional layer and pooling layer of FT d network.

sets contain a series of time-ordered points, each containing latitude, longitude, elevation, and other pieces of information. In the experiments, we use 17621 tracks whose total distance is about 1200000 kilometers, and the total time is more than 48000 hours. These data not only record the user's location at home and at work but also track a wide range of outdoor activities, such as shopping, traveling, hiking, and cycling.

The experimental environment is as follows: Intel(R) Core(TM) i7-4700MQ CPU @ 2.40GHz, with 6G memory.

Programming language was Python.

We use 5s as a time interval to extract trajectory points and then divide these trajectories into sections and each one has 8 points. These sections are real trajectory section set. In addition, we extract and synthesize the algorithm fragment of the current widely used dummy trajectory generation algorithms to generate dummy trajectories. The main steps are shown in Algorithm 3.

We use Algorithm 3 to generate dummy trajectory section set and then put dummy trajectories and real trajectories into CNN for training.

In Table 2, the confusion matrix is listed, which is also known as error matrix. It is a standard format for accuracy evaluation, which is expressed in the form of matrix of N rows and N columns. In AI, confusion matrix is a visualization tool, especially for supervised learning. Unsupervised learning is generally called matching matrix.

Table 2: Confusion matrix.

Predicted value	True value	
	0 (dummy)	1 (real)
0 (dummy)	TN (True Negative)	FN (False Negative)
1 (real)	FP (False Positive)	TP (True Positive)

In our CNN-based dummy trajectory detection scheme, the weight parameter α of RT and FT has a great influence on the detection results. We take seven different values for α and train seven CNN networks. Then we use six different classifier evaluation indicators to evaluate CNN networks. The six evaluation indicators are as follows.

$$\text{Recognition rate} = \frac{TN}{TN + FP}$$

$$\text{Erroneous judgement rate} = \frac{FN}{FN + TP}$$

Input: real trajectory section set
Output: dummy trajectory section set
(1) Procedure
(2) **for** each real trajectory section in real trajectory section set **do**:
(3) **for** i=1 to k **do**:
(4) **for** j=1 to n-1**do**:
(5) randomly selected rotation angle in $(\pi/20, \pi/3) \cup (-\pi/3, -\pi/20)$
(6) Randomly selected expansion rates in(0.5, 1.5)
(7) rotate and expanse trajectory segment j
(8) **end for**
(9) **end for**
(10) **end for**

Algorithm 3: Dummy trajectory section generation.

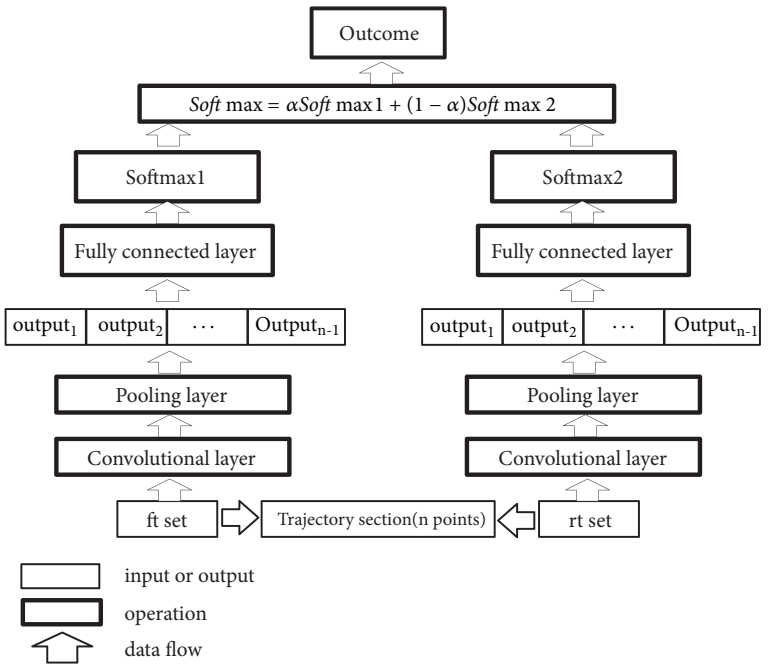


Figure 6: The network architecture of dummy trajectory detection.

$$\begin{aligned} \text{Precision rate P} &= \frac{TP}{TP + FP} \\ \textit{Recall rate } R &= \frac{TP}{TP + FN} \\ \text{F1} - \text{Measure} &= \frac{2 \times P \times R}{P + R} \\ \text{Accuracy} &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{1}$$

Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. That is to say, when the loss function is 0-1 loss, accuracy is the accuracy of the test data set.

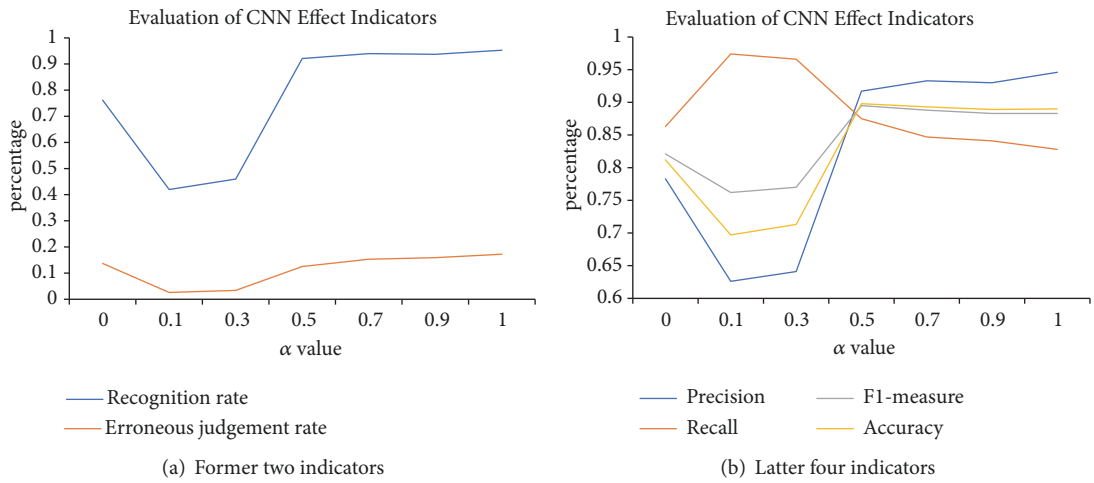
Precision calculates the proportion of items that "should be retrieved" among all items retrieved.

The recall rate calculates the proportion of all items retrieved to all items that should be retrieved.

The comprehensive evaluation index (F1-Measure) is the weighted harmonic average of Precision and Recall. P and R indicators are sometimes contradictory, considering both precision and recall. It is easy to understand that F1 combines the results of P and R. When F1 is higher, the experimental method is more ideal.

For different α values, the values of the six evaluation indicators are shown in Figure 7.

From Figure 7, it can be concluded that the FT attribute of the trajectory section controls the recognition rate; the higher the weight of FT attribute is, the higher the recognition rate is. And the RT attribute of the trajectory section controls the erroneous judgement rate; the higher the weight of RT attribute is, the lower the erroneous judgement rate is. F1-measure and the accuracy are comprehensive global



(a) Former two indicators (b) Latter four indicators

Figure 7: Evaluation indicators of different α value.

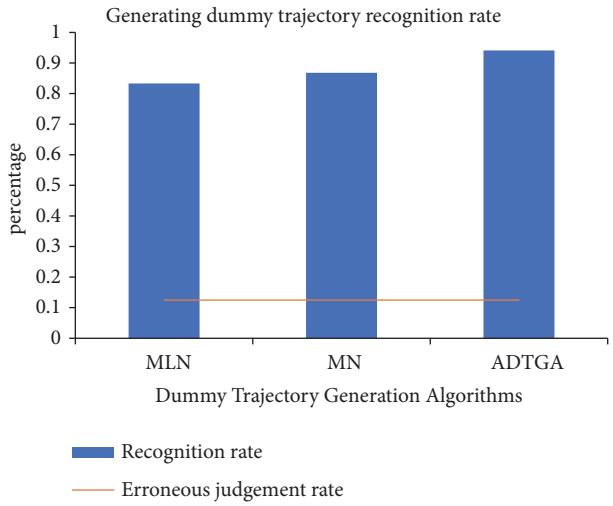


Figure 8: Detection results of dummy trajectory generation algorithms with $\alpha = 0.5$.

evaluation indicators, and the maximum value is obtained when α is 0.5. At the same time, when α is 0.5, the recognition rate is high and the erroneous judgement is low. Therefore, we choose the weight parameter $\alpha = 0.5$.

To verify the efficiency of CNN detection network, three classical algorithms are used to generate dummy trajectories, in which MLN and MN algorithms are proposed in [18], and ADTGA algorithm is proposed by [15]. The detection results are shown in Figure 8.

We use a data set named GeoLife GPS Trajectories as our real trajectory data set and use MLN, MN, and ADTGA to generate large number of dummy trajectories and then randomly select 1000 dummy trajectory sections from the dummy trajectory set and put them into the trained CNN detection network. For MLN, the dummy trajectory recognition rate is 83.3%. For MN, the dummy trajectory recognition rate is 86.8%. And, for ADTGA, the dummy trajectory recognition rate is 94.1%. But the erroneous judgement rate of real trajectories is only 12.5%.

Many dummy trajectory algorithms are improved by the three algorithms mentioned above, for example, literatures [13, 24]. The most significant improvement is the selection of dummy trajectories rather than the generation of dummy trajectory. Therefore, the improved dummy trajectory generation algorithm cannot reduce the detection rate. The experiment illustrates that our dummy trajectory detection scheme based on CNN can detect the dummy trajectory with high recognition rate, while keeping the low erroneous judgement rate. Generally, a complete trajectory has multiple trajectory sections. As shown in Figure 9, for dummy trajectories with multiple trajectory sections, we calculate their recognition rate and the erroneous judgement rate of real trajectories.

With the increase of the number of trajectory sections, the trend of the recognition rate is also increasing, while the erroneous judgement rate shows a downward trend, and the recognition rate is above 90%, and the erroneous rate is below 10%.

5. Conclusion

We have studied many algorithms to generate the dummy trajectories to protect privacy, most of which only take into account the geometric meaning of trajectories without considering the human mobility model. In order to address this weakness, we define two trajectory representation methods and put these two trajectory representation methods of real and dummy trajectories into the improved CNN for training. Experiments show that the deep learning machine CNN is universal; it can identify more than 90% of dummy trajectories that are generated using the current mainly algorithm; meanwhile its erroneous judgement rate is below 10%.

Indeed, our detection scheme cannot be applied to all dummy trajectory generation algorithms. There are two kinds of dummy trajectory generation algorithms; our detection

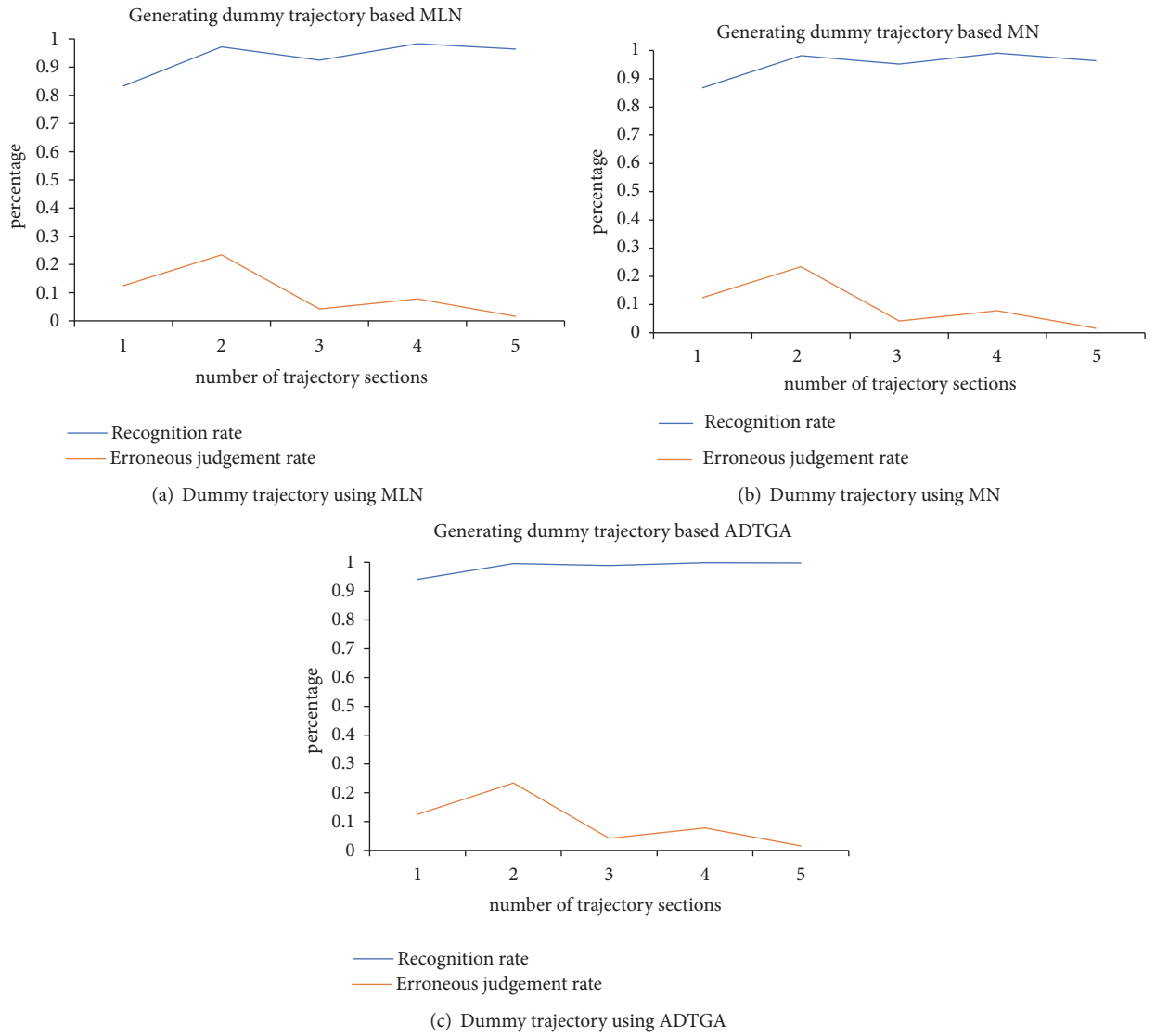


Figure 9: Recognition rate and erroneous rate under the assumption that the discrimination is 0.5.

scheme is powerless. One is simple rotation algorithm and the other is to select similar trajectories or historical trajectories of other users as dummy trajectories. However, the first dummy trajectory generation algorithm is not flexible enough to meet the real background, and the second one needs to collect a large amount of historical trajectory information of the surrounding users, which is very difficult. So these two methods of dummy trajectory generation are difficult to be used in practice. Generally speaking, our dummy trajectory detection scheme has a high detection rate for the dummy trajectory generated by the deformed dummy trajectory generation method.

Our experiments show that the common flaws of the dummy trajectory generation algorithms up till now are that they only consider the geometry of the trajectory points and the trajectory segments and regard them as the products isolated from human behavior and the products of random probability. It is debatable whether such a convolutional neural network learning machine can act as a filter to filter out most of the dummy trajectories which do not conform to reality, as well as leaving behind some dummy trajectories which mix the spurious with the genuine. After all, in the era of exploding CPU and GPU performance, the time cost of generating redundant dummy trajectory sets is negligible.

Data Availability

The data used to support the findings of the manuscript are obtained from Microsoft research GeoLift project; they can be used freely and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Krumm, "A survey of computational location privacy," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, 2009.

[2] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel, "A classification of location privacy attacks and approaches," *Personal and Ubiquitous Computing*, vol. 18, no. 1, pp. 163–175, 2014.

[3] M. Tang, Q. Wu, G. Zhang, L. He, and H. Zhang, "A new scheme of LBS privacy protection," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, pp. 1–6, Beijing, China, September 2009.

[4] W. He, "Research on LBS privacy protection technology in mobile social networks," in *Proceedings of the 2nd IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC '17)*, pp. 73–76, Chongqing, China, March 2017.

[5] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 348–357, ACM, New York, NY, USA, November 2009.

[6] T. Xu and Y. Cai, "Exploring historical location data for anonymity preservation in location-based services," in *Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM '08)*, pp. 547–555, IEEE, Phoenix, AZ, USA, April 2008.

[7] Y. Wang, D. Xu, X. He, C. Zhang, F. Li, and B. Xu, "L2P2: location-aware location privacy protection for location-based services," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 1996–2004, IEEE, Orlando, FL, USA, March 2012.

[8] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating EVES algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, pp. 1–9, 2019.

[9] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 906–912, 2018.

[10] Z. Liu, X. Huang, Z. Hu, M. K. Khan, H. Seo, and L. Zhou, "On emerging family of elliptic curves to secure internet of things: ECC comes of age," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248, 2017.

[11] Y. Liu and Q. Zhao, "E-voting scheme using secret sharing and K-anonymity," *World Wide Web*, pp. 1–11, 2018.

[12] Y. Liu, Y. Wang, X. Wang, Z. Xia, and J. Xu, "Privacy-preserving raw data collection without a trusted authority for IoT," *Computer Networks*, vol. 148, pp. 340–348, 2019.

[13] T.-H. You, W.-C. Peng, and W.-C. Lee, "Protecting moving trajectories with dummies," in *Proceedings of the 8th International Conference on Mobile Data Management (MDM '07)*, pp. 278–282, Mannheim, Germany, May 2007.

[14] P.-R. Lei, W.-C. Peng, I.-J. Su, and C.-P. Chang, "Dummy-based schemes for protecting movement trajectories," *Journal of Information Science and Engineering*, vol. 28, no. 2, pp. 335–350, 2012.

[15] X. Wu and G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW '14)*, pp. 1120–1125, Shenzhen, China, December 2014.

[16] R. Kato, M. Iwata, T. Hara et al., "A dummy-based anonymization method based on user trajectory with pauses," in *Proceedings of the 20th ACM International Conference on Advances in Geographic Information Systems (AGIS '12)*, pp. 249–258, ACM, New York, NY, USA, 2012.

[17] M. Terrovitis and N. Mamoulis, "Privacy preservation in the publication of trajectories," in *Proceedings of the 9th International Conference on Mobile Data Management (MDM '08)*, pp. 65–72, Beijing, China, April 2008.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proceedings of the 2nd International Conference on Pervasive Services (ICPS '05)*, pp. 88–97, IEEE Press, Santorini, Greece, July 2005.

[19] H. Ngo and J. Kim, "Location privacy via differential private perturbation of cloaking area," in *Proceedings of the IEEE 28th Computer Security Foundations Symposium (CSF '15)*, pp. 63–74, Verona, Italy, July 2015.

[20] L. Ou, Z. Qin, Y. Liu, H. Yin, Y. Hu, and H. Chen, "Multi-user location correlation protection with differential privacy," in *Proceedings of the IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS '16)*, pp. 422–429, Wuhan, China, December 2016.

[21] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW '05)*, p. 1248, Tokyo, Japan, April 2005.

[22] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '14)*, pp. 754–762, IEEE, Toronto, Canada, April-May 2014.

[23] T. Hara, Y. Arase, A. Yamamoto, X. Xie, M. Iwata, and S. Nishio, "Location anonymization using real car trace data for location based services," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication (ICUIMC '14)*, pp. 1–8, ACM, New York, NY, USA, January 2014.

[24] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 16–23, ACM, New York, NY, USA, June 2008.

[25] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proceedings of the 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS '10)*, pp. 398–401, ACM, November 2010.

[26] R. Kato, M. Iwata, T. Hara, Y. Arase, X. Xie, and S. Nishio, "User location anonymization method for wide distribution of dummies," in *Database and Expert Systems Applications*, vol. 8056 of *Lecture Notes in Computer Science*, pp. 259–273, Springer, Berlin Heidelberg, 2013.

[27] R. Lan, J. He, S. Wang, Y. Liu, and X. Luo, "A parameter-selection-based chaotic system," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 3, pp. 492–496, 2019.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] L. Bottou, Y. Bengio, and Y. Le Cun, "Global training of document processing systems using graph transformer networks,"

in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 489–494, San Juan, PR, USA, June 1997.

[30] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '14)*, pp. 1746–1751, Doha, Qatar, October 2014.