TSHN: A Trajectory Similarity Hybrid Networks for Dummy Trajectory Identification

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Nowadays, people pay more and more attention to privacy protection with the continuous occurrence of data breaches. In location-based services, dummy trajectory generation is a popular location privacy protection method. However, this method is used by some malicious users for benefits, which results in economic losses and the waste of resources of the location-based services provider. For this problem, dummy trajectory identification has been proposed by researchers. Nevertheless, with the continuous development of dummy trajectory generation algorithms, the existing dummy trajectory identification methods are unsuitable. In this paper, we propose a hybrid neural network framework for dummy trajectory identification, called trajectory similarity hybrid networks (TSHN). The main idea of TSHN is to identify whether a target trajectory is a virtual trajectory according to the similarity score between historical trajectories and the target trajectory. For each historical trajectory of the user, the mobility and individual features are extracted to train TSHN. The trajectory similarity score generated by the TSHN is used to identify dummy trajectories. The experimental results show that our proposed TSHN can identify the dummy trajectory with an accuracy of 0.97, which significantly outperforms the existing dummy trajectory identification methods.**

***Index Terms*—Dummy trajectory identification, Neural networks, location privacy, mobility feature, individual feature.**

I. Introduction

In recent years, with the continuous occurrence of data breaches, people pay more and more attention to privacy protection [1]. In the scenario of location-based services (LBSs), dummy-based method is a popular location privacy protection technique that conceals the real request among fake requests. To protect the location privacy when users continuously request the LBS, various dummy trajectory generation methods have been proposed. However, this method is maliciously used by some malicious users in the real world for benefit, and Fig. 1 shows two examples of the misuse of this method. In example 1, the malicious user gets the preferential qualification by changing the mobile phone's GPS location, which may cause economic losses to the location-based service provider (LSP) [2], [3]. As a real-world example, some hackers have published the method of getting discounts on the Meituan app by using dummy location[1]. In example 2, the user uses the dummy location for remote check-in, i.e., the user changes the GPS location on the mobile phone to the company's location for check-in at home. This will cause economic losses to the company and is illegal in China[2]. These examples demonstrate that the dummy trajectory identification technique is necessary for the company and LSP, which can detect the dummy (fake) location.

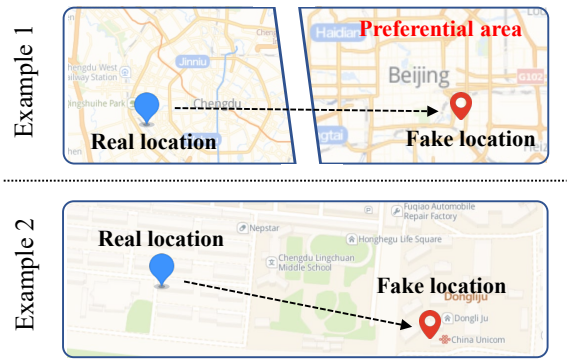


Fig. 1. Two examples of the misuse of fake locations.

Some previous works similar to the dummy trajectory recognition have focused on user identification based on trajectories. Hallac et al. [4] presented an identity prediction approach for drivers based on automobile sensor data. They collected the dataset by sensors and focused on the 12 most frequent turns. Chowdhury et al. [5] proposed a driver identification approach using only raw GPS data. They extracted a set of 137 statistical features from all trajectories and concluded that a random forest classifier is best for classifying trajectories. Kieu et al. [6] presented a deep learning model based on a convolutional neural network (CNN) to distinguish trajectories from different drivers, which transformed the trajectory into a 3D matrix. Ren et al. [7] defined the problem of human mobility signature identification and proposed a spatio-temporal siamese network to address this problem by calculating the similarity score of the trajectories. Pan et al. [8] proposed a dummy trajectory detection algorithm based on CNN, which utilized the direction and location features of the trajectory.

Nonetheless, there are still some problems in applying these techniques for dummy trajectory identification. First, some of

[1]https://www.onezyh.cn/1652.html

[2]http://www.gov.cn/zhengce/2020-12/25/content_5575080.htm

these techniques need to install sensors on the transportation to collect additional data, e.g., Hallac et al.'s method [4]. However, in the real world, the cost of installing sensors is high for all vehicles, and it is inconvenient for most people to install sensors. Second, most of these works are only implemented on the driving trajectory, e.g., [4]–[7]. However, in real-world situations, there are different types of trajectories, e.g., walk, train, and bus, these driver identification frameworks through driving trajectories may not be applicable to these situations. Third, some previous works do not take into account the individual features extracted from the trajectories [4]–[6], [8]. However, the individual feature reflects the semantic information of the real trajectories, which is significant for the dummy trajectory identification.

In this paper, we present a novel trajectory similarity hybrid networks (TSHN) framework to identify the dummy trajectory using historical trajectory data to address these challenges. Specifically, an improved Trajectory-to-Image (iT2I) encoding scheme [6] is proposed firstly to transform every trajectory into a 3D matrix for mobility features extraction. The 3D matrix contains five $256 * 256$ feature matrices, that capture the shape features, time differences, speeds, accelerations, and directions of trajectory, respectively. Secondly, the individual features are extracted from the historical trajectory data. Thirdly, the 3D matrix and vector of the features are input into TSHN and trained to calculate the similarity score of each pair of trajectories. The experimental results demonstrate that TSHN outperforms all baseline algorithms in dummy trajectory identification ability.

The contributions of this paper are summarized as follows:

- We propose an improved Trajectory-to-Image (iT2I) encoding scheme, which can capture additional information by the time difference matrix compared to T2I.
- We design a novel TSHN framework that use mobility and individual features for model training, and identifies dummy trajectories based on the trajectory similarity score. The source code has been released to Github[3]
- The experimental results demonstrate that our proposed TSHN outperforms all baselines in dummy trajectory identification performances.

The remainder of the paper is organized as follows. Section II introduces some preliminaries used in the paper. Section III introduces TSHN in detail. The experimental and comparison results are given in Section IV, and the related works are introduced in Section V. Finally, Section VI concludes the full paper.

II. Preliminaries

This section first defines the problem formally, and then introduces the used data and the framework of TSHN.

A. Problem definition

Definition 1: GPS point and GPS trajectory. In general, the GPS point is generated by smart devices with a GPS module, such as smartphones and smart watches. A GPS point represents a location with the form of $p = \langle lat, lng, t \rangle$, where lat, lng, and t represent the latitude, the longitude, and the timestamp of the GPS point, respectively. Correspondingly, a trajectory $traj$ consists of a set of GPS points ordered by timestamp, denoted as $traj = \langle p_1, p_2, \ldots, p_n \rangle$, where $p_i.t \leq p_{i+1}.t$ and $i \in [1, n-1]$. A user's historical trajectories are denoted as $\mathcal{DB} = \{traj_1, traj_2, \ldots, traj_n\}$.

Definition 2: Points of interest (POI). A point of interest (POI) is a specific location point that may useful or interesting to someone. In general, it contains not only information about coordinates but also additional information about the location, such as name, address, and type. In this paper, the form of a POI is $poi = \langle lat, lng, type \rangle$, where lat, lng, and $type$ represent the latitude, the longitude, and type of POI, respectively. Each trajectory contains at least two POIs, namely poi_{st} and poi_{en}, which represent the origin and destination, respectively.

Definition 3: Mobility features M. Mobility features represent a user's behavior patterns that are derived from the user's specific habits or skills, such as speed, acceleration, and turning location. For example, some users are used to driving at high speeds, and some users know how to get from home to the workplace quickly. To more easily describe the user's mobility behavior, we transform each trajectory into a 3D matrix M_i and utilize the CNN to dig deep features, where i is the ID of the trajectory.

Definition 4: Individual features f_{ind}. Each user has his/her own unique individual (personal or profile) features that can be extracted from historical trajectories $\mathcal{DB}$, e.g., points of interest, the start/end times, trip duration and distance of the trip. In the experiments, for each trajectory of the user, we extract a five-dimensional feature vector f_{ind} and denote the i-th dimension of individual feature as $f_{ind,i}$.

Problem definition Given a user's historical trajectory dataset $\mathcal{DB}$, and a target trajectory $traj_x$ that may be a dummy trajectory or not. The problem to be solved in this paper is to judge whether the target trajectory $traj_x$ is a dummy trajectory for the given $\mathcal{DB}$. We aim to design a model TSHN to solve this problem.

B. Data Description

In order to illustrate the effectiveness of TSHN, real-world datasets are applied to this framework. The input data mainly contains 3 types of data, i.e., (1) historical trajectory data $\mathcal{DB}$, (2) dummy trajectory data, and (3) map data.

Historical trajectory data is taken from a public dataset in our work, i.e., Geolife v1.3. The Geolife trajectory dataset was collected by 182 users in Beijing, China, from 2007 to 2012. Overall, 17621 trajectories were collected with a total duration of 50176 hours. Each trajectory consists of a sequence of GPS points, and each GPS point contains 7 key data fields, including timestamp, latitude, longitude, etc.

Dummy trajectory data is generated by some dummy trajectory generation algorithms, such as random, MN, MLN, and ADTGA [9], [10]. For consistency, the dummy trajectory is

[3]https://github.com/fang-zhiyou/TSHN

also located in Beijing, and the number of dummy trajectories is equal to that of the historical trajectory data.

Map data of Beijing covers the area between 38.86° to 40.02° in latitude and 116.2° to 116.48° in longitude.

C. Solution Framework

The primary function of our proposed TSHN framework is to identify dummy trajectories according to a user's historical trajectories. The workflow of solution framework is shown in Fig. 2, which consists of 2 parts:

1) ***Feature Extraction****:* According to the $\mathcal{DB}$, the individual features and mobility features of the trajectory are extracted, which can be seen in the upper part of Fig. 2.

2) ***Learning Model****:* The bottom half of Fig. 2 shows the details of the TSHN framework, which takes above two types of features as input and outputs the result of similarity score between two trajectories.

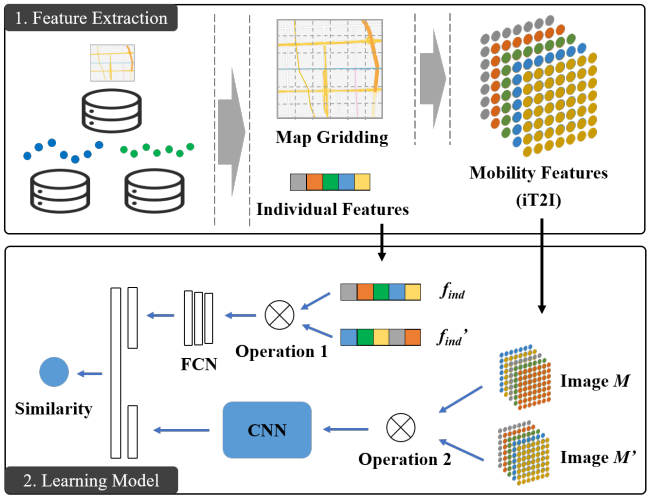


Fig. 2. The workflow of solution framework.

III. Proposed Solution: TSHN

In this section, the TSHN is described in detail. The mobility and individual features are introduced first according to the GPS trajectory and map data. Then, the learning model is introduced.

A. Mobility Features Extraction

To better represent the mobility features of the trajectory, it is crucial to utilize as much information about trajectories as possible. In general, a trajectory implies three different types of features: (1) geographic features, which are expressed in longitude and latitude. (2) behavior features, such as speed and directions. (3) union features, such as the reachability of the user's trajectory in space and time. In order to distinguish between dummy trajectories and real trajectories, it is essential to represent trajectories based on these features.

In order to achieve the above goal, the Trajectory-to-Image (T2I) encoding scheme [6] is adopted to represent a trajectory. However, although the T2I scheme can represent geographic and behavior features by transforming a trajectory into a 3D matrix, the raw T2I is insufficient to represent the mobility features, especially union features. This is because the T2I does not consider the time attribute of the trajectory, which can reflect some important features of the trajectory, such as the habit of the user and union features. For example, the T2I cannot reflect a user's habit of going to school at 8 o'clock every day. To overcome this shortcoming, an improved T2I (iT2I) is proposed by adding a time difference matrix based on T2I. The details of iT2I are described below:

1) Map Gridding: Generally, the real GPS coordinates are not 100% accurate, and the real trajectories formed by the real GPS coordinates are not smooth. Therefore, the griding technique [6], [7], [11]–[13] with high scalability in practice is used to represent a trajectory for accuracy. Specifically, the map is divided into $\alpha \times \beta$ equal-sized grid cells, where a grid cell is denoted as $[x, y]$, $(0 < x \leq \alpha, 0 < y \leq \beta)$. For example, in the left part of Fig. 3, an area of Beijing is divided into 13×16 grid cells, where the continuous blue icons represent a trajectory consisting of several GPS points.

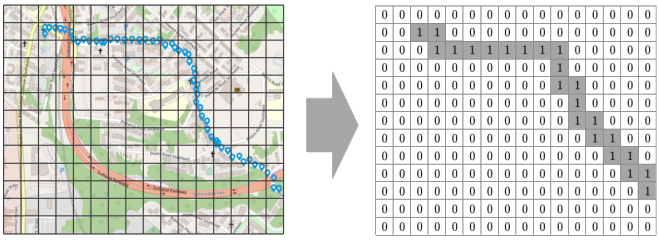


Fig. 3. The Generation Process of $M_{i,1}$.

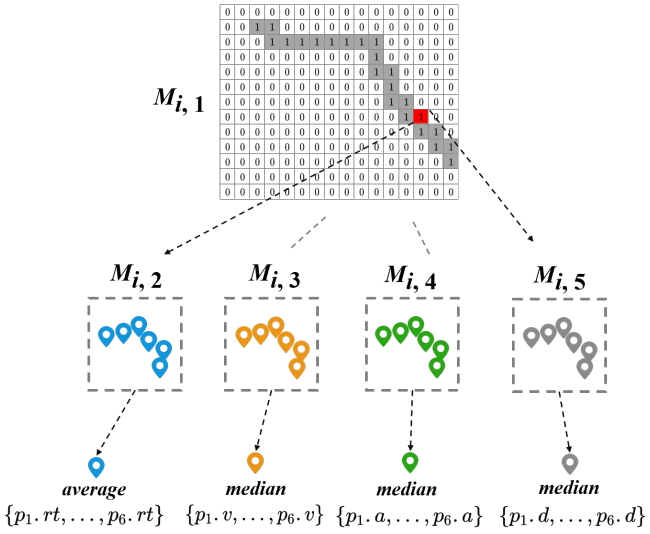


Fig. 4. The Generation Process of $M_{i,2}$, $M_{i,3}$, $M_{i,4}$, and $M_{i,5}$.

2) 3D matrix construction: Based on the map gridding, the five matrices are constructed, i.e., $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, $M_{i,4}$, and $M_{i,5} \in \mathbb{R}^{\alpha \times \beta}$, where $M_{i,j}$ represents the jth matrix of the ith trajectory in $\mathcal{DB}$ and $1 \leq j \leq 5$. $M_{i,1}$ captures the geographic features by trajectory encoding. $M_{i,2}$ captures the user's union

features by the relative time difference. $M_{i,3}$, $M_{i,4}$, and $M_{i,5}$ capture the speeds, directions, and accelerations of the trajectory, respectively. Besides, the size of $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, $M_{i,4}$, and $M_{i,5}$ are the same for consistency. The detailed construction process of each matrix is as follows:

For $M_{i,1}$, the element $M_{i,1}[x, y]$ is set to 1, if at least one GPS point of the trajectory M_i is located in cell $[x, y]$. Otherwise, the $M_{i,1}[x, y]$ is set to 0. The right part of Fig. 3 shows the transformation result of $M_{i,1}$. For example, $[2, 3]$ and $[1, 1]$ are set to 1 and 0, respectively, because there are GPS points of the trajectory M_i in $[2, 3]$ but not in $[1, 1]$.

To construct $M_{i,2}$, which captures the union features, the relative time rt of each GPS point is set to the timestamp minus date. For example, if the timestamp of a GPS point p is "2022-06-20 08:08:08", p's relative time $p.rt$ is 29340, i.e., $8 * 3600 + 8 * 60 + 8$. So, the element $M_{i,2}[x, y]$ is set to the average of the relative times of all GPS points in $[x, y]$.

Since the construction processes of $M_{i,3}$, $M_{i,4}$, and $M_{i,5}$ are similar, in the following, we only describe the construction of the $M_{i,3}$, which captures the speed features.

To construct $M_{i,3}$, the median speed is used to set the value of the matrix. Specifically, for a trajectory $traj = \langle p_1, p_2, \cdots, p_n \rangle$, the speed sequence $sp\_seq = \{v_1, v_2, \ldots, v_n\}$ is first calculated by equation $v_i = \frac{dis(p_i, p_{i+1})}{p_{i+1}.t - p_i.t}$ and $v_1 = 0$, where dis is the Euclidean distance function. Then, if several GPS points of the trajectory are located in the cell $[x, y]$, the value of the $M_{i,3}[x, y]$ is set to the median speed of the speed sequence produced by these GPS points. Since the speed is calculated by dividing the straight-line distance between two GPS points by the time, the result is lower than the real speed. So, the median speed is used instead of the average speed [14]. For example, there are 7 GPS points in the cell $[2, 3]$, and the corresponding speed sequence is $sp\_seq = \{17, 16, 18, 5, 16, 17, 18\}$, which may represent a scenario where the user waits a traffic light and passes a corner. Since, the median speed $med\_speed = 17$, the $M_{i,3}[2, 3]$ is set to 17. While the $ave\_speed \simeq 15.3$, is influenced by the low speed in the corner. An example is shown in Fig. 5.

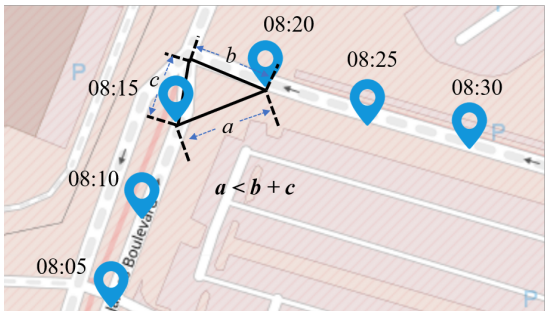


Fig. 5. An example of Corner Error

After that, the trajectory is transformed into $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, $M_{i,4}$, and $M_{i,5}$. As shown in Fig. 4, these matrices are transformed into a 3D matrix $M_{i,3D} \in \mathbb{R}^{\alpha \times \beta \times 5}$, which is viewed as a 5 channels image.

B. Individual Features Extraction

Generally, each user has unique individual (profile or personal) characteristics, such as points of interest, life schedule, etc. In this paper, five individual features are extracted from each trajectory by existing algorithms [15].

$f_{ind,1}$: The coordinates (longitude and latitude) of the POI. The user has his/her own POI, where to have lunch or take a break. Since the map griding is used in our work, a POI can be represented by the corresponding cell $[x, y]$.

$f_{ind,2}$&$f_{ind,3}$: The distance and time of the trajectory. In the real world, different users have different hobbies, some users prefer long trips and some prefer short trips. Since the distance and time of the trajectory can reflect the user's habits about the trips, thus, we extract these features of the trajectory.

$f_{ind,4}$: Start time. The user has his/her own favorite time to travel. For example, if the user usually walks at night, the trajectory with the start time in the daytime may be a dummy. The start time can capture the user's schedule, which is critical in identifying the dummy trajectory.

$f_{ind,5}$: The average speed of the trajectory. The speed is an essential element in identifying the dummy trajectory as it generally remains the same and reflects the travel mode, such as walking and driving. So, we extract the average speed as an individual feature.

Based on the above analysis, the individual features can be represented by a vector of length 5.

C. Learning Model

More and more mobile devices with GPS modules accumulate a large amount of GPS data, making it possible to identify dummy trajectories from large-scale trajectories. However, there are two challenges in realizing this goal. (1) How to capture trajectory similarity information from the mobility or individual features. (2) How to combine mobility and individual features, i.e., a 3D matrix and a vector. For the first challenge, we propose the MatchPyramid-based convolutional neural network (MPCNN) and the MatchPyramid-based fully connected network (MPFC) to capture the similarity information from 3D matrices and feature vectors, respectively. For the second challenge, we propose a Trajectory Similarity Hybrid Networks (TSHN) for dummy trajectory identification by combing the MPCNN and MPFC and adding a combing fully connected network (CFC) which can combine the results of MPCNN and MPFC.

The architecture of the TSHN is shown in Fig. 6, which consists of MPCNN, MPFC, and CFC. The detailed description of the model training for dummy trajectory identification is as follows:

(1) MatchPyramid-based convolutional neural network (MPCNN). For the first challenge, based on the MatchPyramid architecture [16], we present MPCNN, which can take two multi-channel images as input and capture the similarity information between two images. The details of the MPCNN are described below:

MPCNN is shown in the top left of Fig. 6. For any two trajectories (A and B), in order to calculate their similarity

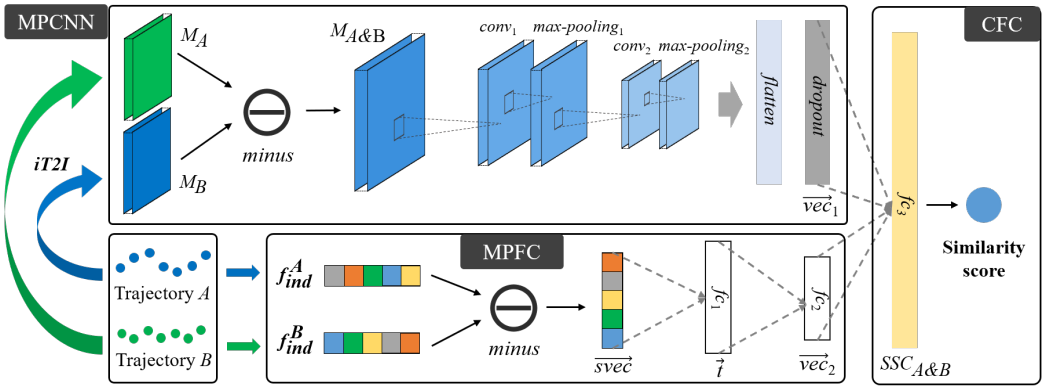


Fig. 6. The Architecture of TSHN

score SSC, iT2i (Section III-A) is first used to encode them as two 3D matrices, i.e., M_A and M_B. Then, the minus operation $\ominus$ is performed on them and obtains a 3D matrix $M_{A\&B}$. Next, $M_{A\&B}$ (as the input) is processed by a CNN, which consists of two convolutional layers and two max-pooling layers, as shown in Fig. 6.

In the first convolution layer, the similarity information is learned by convolving the 3D matrix $M_{A\&B}$ using Equation 1 with 5 different filters.

$$z^k_{A\&B} = \sigma(M_{A\&B} \otimes W^k + b^k), 1 \leq k \leq 5 \tag{1}$$

Here, $z^k_{A\&B}$ is the output of the first convolution layer and is a matrix called the k-th features map of $M_{A\&B}$. In the right part of Equation 1, W^k is the k-th filter ($1 \leq k \leq 5$), which is initialized using Kaiming Initialization and the size of the filter is $3 * 3$; b^k is a bias, which is initialized by 0; symbol $\otimes$ represents the convolution operation; and the σ is the Rectified Linear Unit (ReLU) function (activation function).

In general, a convolutional layer is followed by a pooling layer, which is a sampling method. The main function of the pooling layer is to retain the main information while reducing the number of parameters and calculations to prevent overfitting. As shown in Fig. 6, the max-pooling technique is used instead of the mean-pooling technique because it can learn the edge and texture structure of the image.

The next convolutional and max-pooling layers are applied similarly. After that, the matrix is flattened into a long vector and connected to a dropout layer [17], which can overcome overfitting. The output of the MPCNN is a vector, denoted as $\overrightarrow{vec}_1$.

(2) MatchPyramid-based fully connected network (MPFC). For the individual features f^A_{ind} and f^B_{ind}, the MPFC is proposed to capture the similarity information between them. The details of MPFC are described below.

MPFC is shown in the lower left of Fig. 6, which is used to process the individual features f^A_{ind} and f^B_{ind}. The MPFC contains one operation and two FC layers, i.e., $\ominus$, fc_1, and fc_2. For any two individual features from different trajectories, the $\ominus$ operation is performed on them to calculate a new similarity vector $\overrightarrow{svec}$. The function of fc_1 is to capture similar features of two trajectories, which works as a bottleneck [18]. fc_2 is used to construct/reconstruct vector $\overrightarrow{vec}_2$, i.e., the output of the MPFC. The $\ominus$ operation is minus and the activation function of the fc_1 and fc_2 is $sigmod$, so we have:

$$\vec{t} = sigmod(\overrightarrow{svec} \cdot w_1 + b_1) \tag{2}$$

$$\overrightarrow{vec}_2 = sigmod(\vec{t} \cdot w_2 + b_2) \tag{3}$$

where $\vec{t}$ is the output of fc_1, w_1 and w_2 are weighting matrices, and b_1 and b_2 are bias matrices.

(3) Combing fully connected network (CFC). The function of the CFC is to calculate the similarity score. It first aggregates the outputs of MPCNN and MPFC to get $\overrightarrow{vec}_3 = \overrightarrow{vec}_1 \| \overrightarrow{vec}_2$. Then, $\overrightarrow{vec}_3$ is connected to layer fc_3 to calculate the similarity score as Equation 4. The activation function of the fc_3 is also $sigmod$.

$$SSC_{A\&B} = sigmod((\overrightarrow{vec}_1 \| \overrightarrow{vec}_2) \cdot w_3 + b_3) \tag{4}$$

where w_3 and b_3 are a weighting vector and a bias vector, respectively. $SSC_{A\&B}$ is the output of fc_3, which represents the similarity score of the two trajectories. In our work, trajectories A and B are similar if their SSC is less than 50.

IV. Experimental Study

In this section, the Geolife dataset and the generated dummy trajectories are used to test effectiveness of the proposed TSHN framework. We also compare TSHN with other baseline methods, such as SVM, FC, and MatchPyramid [8], [19].

A. Experimental Setup

In the experiment, 100 real trajectories $\langle r_1, r_2, \ldots, r_{100} \rangle$ and 100 dummy trajectories $\langle d_1, d_2, \ldots, d_{100} \rangle$ are used to construct the training and testing datasets. The 100 real trajectories are chosen from the user 128 in the Geolife dataset. The 100 dummy trajectories are generated by random algorithm, MN, MLN, or ADTGA algorithms [9], [10]. Some definitions about the experimental setup are described below.

Item: $\langle A, B, score \rangle$, A record of the similarity of two trajectories, where A, B are trajectories, and $score$ represents their similarity score.

Training dataset: We utilize 80 real trajectories and 80 dummy trajectories to construct training dataset, which consists of 3160 100-score items and 3160 0-score items. The 3160 100-score items is $\{\langle r_1, r_2, 100\rangle, \ldots, \langle r_{79}, r_{80}, 100\rangle\}$, where r_i is real trajectory, and 3160 0-score items is $\{\langle r_1, d_2, 0\rangle, \ldots, \langle r_{79}, d_{80}, 0\rangle\}$, where r_i is real and d_i is dummy.

Testing dataset: We utilize 20 real trajectories and 20 dummy trajectories to construct the testing dataset which consists of 190 100-score items and 190 0-score items. The 190 100-score items $\{\langle r_1, r_2, 100\rangle, \ldots, \langle r_{19}, r_{20}, 100\rangle\}$, where r_i is real trajectory, and the 190 0-score items $\{\langle r_1, d_1, 0\rangle, \ldots, \langle r_{19}, d_{20}, 0\rangle\}$, where r_i is real and d_i is dummy.

Evaluation metrics: Given a target trajectory $traj_x$, the similarity score between $traj_x$ and every real trajectory trained in TSHN is calculated and a vector $SSC = \{ssc_1, ssc_2, \ldots, ssc_{80}\}$ is generated. Then, the mean score $score$ of SSC is calculated. If the $score \leq 50$, the target trajectory $traj_x$ is identified as a dummy trajectory. Otherwise, $traj_x$ is identified as a real trajectory. Besides, the indexes of accuracy, precision, and F_1 score are used to evaluate the performance of our proposed TSHN and baseline methods.

B. Baseline Algorithms

We compare the performances of our proposed TSHN with the following baseline algorithms:

(1) SVM. A linear SVM is used to test the similarity of faces [19]. In the experiment, SVM is used as a baseline algorithm, and we only take the individual features as input to identify dummy trajectory.

(2) Fully connected neural network (FC): Fully connected neural network is the basic classification or regression model in deep learning. We use the MPFC to identify the dummy trajectory. Thus, the result of the output is only based on individual features.

(3) Naive MatchPyramid. In previous work [8], MatchPyramid is used to calculate the similarity score of two sentences, which can be represented by the vector. So, we use the Naive MatchPyramid to identify the dummy trajectory, and the similarity score is based on individual features.

C. Experimental Results

TABLE I
MODEL COMPARISON

Schemes	Accuracy	Precision	F_1 score
TSHN	**0.9700**	0.9700	**0.9700**
SVM	0.9250	**1.0000**	0.9189
Naive MatchPyramid	0.7150	0.7450	0.7600
MPFC (a part of TSHN)	0.7400	0.7500	0.7700
MPCNN (a part of TSHN)	0.7750	0.6900	0.8200
TSHN (T2I)	0.9000	0.8900	0.9250
MPCNN (T2I)	0.7200	0.7300	0.7500

*1) **Vertical comparison**:* The experimental results of our proposed TSHN and the baseline algorithms are shown in table I. The accuracy, precision, and F_1 score are used to compare these algorithms. The TSHN, TSHN(T2I), MPCNN, MPCNN(T2I) and MPFC are trained by the training dataset described above. Meanwhile, the SVM and MatchPyramid algorithms are trained by raw 100 real trajectories and 100 dummy trajectories. Table I demonstrates that our proposed TSHN has the best performance to identify dummy trajectory. It is noted that SVM outperforms other baseline algorithms but is worse than TSHN. The main reason is that the SVM only uses the individual features without mobility features. Furthermore, we compare our TSHN with MPCNN and MPFC, which capture one feature of the trajectory separately, i.e., MPCNN uses 3D matrix to capture mobility features based on CNN, while MPFC uses FC to capture individual features. The experimental results show that both individual features and mobility features provide some useful information for dummy trajectory identification, but the effect of using one feature alone for dummy trajectory identification is far worse than combining two features. Besides, we also test the effect of TSHN(T2I) and MPCNN(T2I). The experimental results show that the iT2I encoding scheme is better than T2I, and this is because the user's trajectory is distributed according to time, which can reflect the user's living habits. Table I shows that the F_1 score and accuracy of the TSHN are over 0.95, which is significantly bigger than others.

*2) **Horizontal comparison**:* In this part, we test the accuracy of THSN to identify dummy trajectories generated by different dummy trajectory generation algorithms, such as random algorithm, MN, MLN, and ADTGA [8], [19]. The experimental results shown in Fig. 7 illustrate that our proposed TSHN works well for all dummy trajectory generation algorithms, i.e., TSHN can identify dummy trajectories generated by advanced virtual trajectory generation algorithms with a recognition rate of over 85%. It is worth noting that although the dummy trajectories generated by ADTGA are similar to real trajectories, the recognition rate of TSHN still reaches 87.5%.

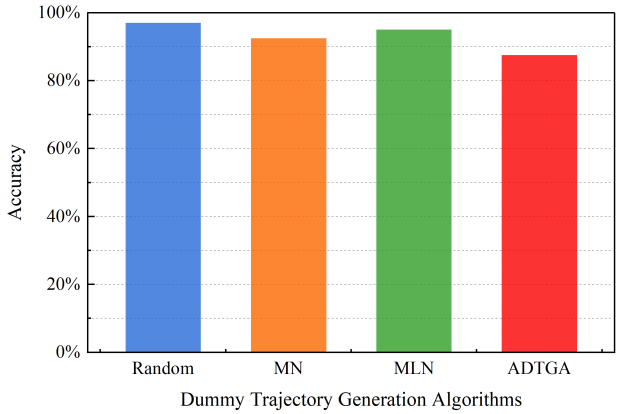


Fig. 7. Identification results of dummy trajectory generation algorithms.

D. Features Analysis

*1) **Importance of the mobility features**:* Mobility features consist of 5 aspects, i.e., geography, speed, acceleration, time difference, and direction. In general, different users have different trajectory distributions and different behavior patterns, which can be reflected by these mobility features. Part of mobility features have been used in previous work [8] to identify dummy trajectories. In this section, we test the effect of mobility features for dummy trajectory identification. As shown in Fig. 8(a), when when only using mobility features for dummy trajectory identification, the accuracy can achieve 77%, which indicates that extracting the mobility features of the trajectory is useful for dummy trajectory recognition. Besides, the recall rate of the MPCNN is 100%, which represents that the mobility features can help the model to identify the real trajectory.

*2) **Importance of the individual features**:* Individual features also consist of 5 aspects, i.e., POI, distance, time, start time, and average speed of the trajectory. Similar to mobility features, we evaluate the importance of individual features in this part. Since the individual features are the obvious characteristics of the user, the individual features are the most common features used in dummy trajectory identification [20]–[22]. As shown in Fig. 8(b), the dummy trajectory recognition accuracy when using only MPFC reaches 73%, which indicates that individual features are helpful for the dummy trajectory identification. Besides, the accuracy, recall, precision, and F_1 score of the MPFC are all higher than 70%.

*3) **TSHN analysis**:* The TSHN framework mainly consists of two different models, i.e., MPCNN and MPFC, and combines mobility and individual features to identify dummy trajectories. As shown in Fig. 8(c), the accuracy of TSHN in identifying dummy trajectories is 97%, which is much higher than that of using MPCNN and MPFC algorithms alone. The experimental result demonstrates that the combination of mobility and individual features is reasonable. Besides, since the parameters of the TSHN are initialized randomly, the performance of the TSHN is not stable. Fig. 8(d) shows the relationship between accuracy and loss of the TSHN, which illustrates that the smaller the loss, the higher the accuracy of the TSHN model.

V. Related Work

To protect location privacy, researchers have proposed dummy trajectory generation algorithms. Meanwhile, in order to prevent the abuse of the location privacy protection based on dummy trajectory generation, researchers have proposed dummy trajectory recognition algorithms. Next, we review related works from these two aspects.

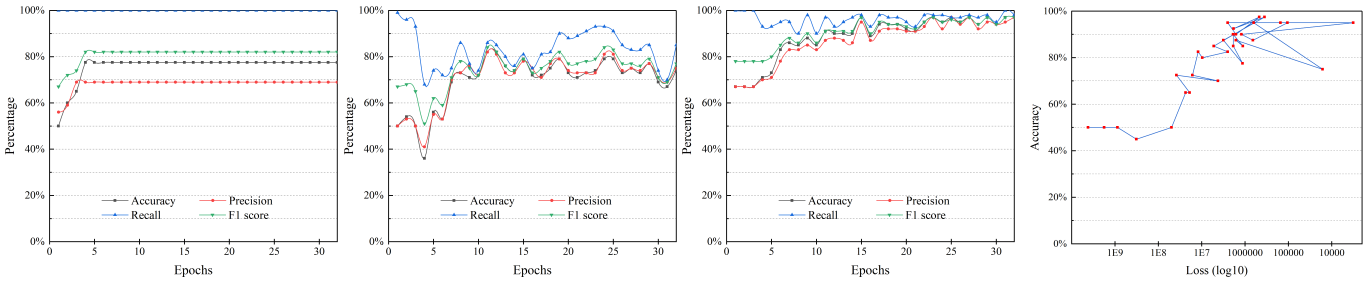
Dummy trajectory generation: Dummy trajectory generation is a popular location privacy protection method, which protects the user's location privacy by generating some dummy trajectories that are indistinguishable from the real trajectories [23]–[26]. Tu et al. [23] considered the semantic information and proposed a dummy trajectory generation algorithm by merging different trajectories, which maintains the high usability of trajectories. Kang et al. [24] proposed an online location privacy protection system that can generate different dummy trajectories based on the moving pattern and profile of the user. According to the point of interest perturbation and overlapping n-grams of trajectory data, Cunningham et al. [25] proposed a local differential privacy-based trajectory generation algorithm based on trajectory sharing. Tang et al. [26] proposed a dummy-based location privacy-preserving scheme by exploiting the deceptive dummy techniques, and the scheme was developed on the Android platform and tested in the real environment.

Dummy trajectory identification: There are few dummy trajectory identification works in recent years, such as [8], [20]–[22]. Most existing works rely on common mobility features of the trajectory, such as speed and direction. Lei et al. [20] shown that sequential pattern mining technique can be used for identify user trajectory patterns. Pan et al. [8] proposed a novel dummy trajectory detection framework based on convolutional neural networks with considering the direction and coordinates of the trajectory. Qian et al. [21] proposed an online abnormal taxi trajectory detection method considering spatio-temporal relations, which improves precision and efficiency. Chen et al. [22] proposed a multidimensional criteria based anomalous trajectory detection method to identify anomalous taxi trajectories online, which reduced the false positives without sacrificing false negative rates.

Trajectory behavior learning: Most existing works on trajectory behavior learning also rely on human-defined mobility features. These mobility features are derived from real-world GPS data [5]–[7], [27], [28]. Supervised learning, unsupervised learning, or reinforcement learning are used by the researchers to solve this problem [6], [7], [27], [29]. Dabiri et al. [27] utilized CNN architectures to identify trajectory modes from raw GPS trajectories, which achieves high accuracy. Dabiri et al. [29] proposed a deep CNN for vehicle classification, which captures the vehicle's behavior from large-scale GPS trajectory data based on their new proposed representation of GPS trajectories. Ren et al. [7] proposed spatio-temporal siamese networks to learn the user's behavior features and identify the identity of the user. Kieu et al. [6] first presented a trajectory-to-image encoding scheme to encode the image into a 3D matrix and proposed a multi-task deep learning model to learn the user's behavior from trajectories.

VI. Conclusion

In this paper, we propose an improved trajectory-to-image encoding scheme (iT2I) by adding a new time difference matrix to capture more mobility information about the trajectory. Next, we design a novel trajectory similarity hybrid network (TSHN) for dummy trajectory identification. The TSHN consists of three parts, i.e., MPCNN, MPFC, and CFC, and combines the mobility and individual features to calculate the similarity score for each pair of trajectories. According to the similarity score, a dummy trajectory can be identified. The experimental results show that our proposed TSHN outperforms



(a) MPCNN (b) MPFC (c) TSHN (d) Loss and Accuracy

Fig. 8. Features Analysis

all baseline algorithms in dummy trajectory identification. In the future, we will use more deep learning algorithms to improve accuracy and test the more state-of-the-art dummy trajectory generation algorithms in TSHN.

REFERENCES

[1] C. Cadwalladr and E. Graham-Harrison, "Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach," *The guardian*, vol. 17, p. 22, 2018.

[2] Z. Jia, "Pricing strategies on online take-out oligopoly marketing: A case study of ele. me," *Scientific Journal of Economics and Management Research Volume*, vol. 4, no. 1, 2022.

[3] X. Ge, "Research on the social problems caused by meituan and the feasibility of adjusting the business model," in *2022 7th International Conference on Social Sciences and Economic Development (ICSSED 2022)*. Atlantis Press, 2022, pp. 2134–2138.

[4] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, M. Roehder, J. Leskovec *et al.*, "Driver identification using automobile sensor data from a single turn," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 953–958.

[5] A. Chowdhury, T. Chakravarty, A. Ghose, T. Banerjee, and P. Balamuralidhar, "Investigations on driver unique identification from smartphone's gps data alone," *Journal of Advanced Transportation*, vol. 2018, 2018.

[6] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Distinguishing trajectories from different drivers using incompletely labeled trajectories," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, ser. CIKM '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 863–872.

[7] H. Ren, M. Pan, Y. Li, X. Zhou, and J. Luo, "St-siamesenet: Spatio-temporal siamese networks for human mobility signature identification," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1306–1315.

[8] J. Pan, Y. Liu, and W. Zhang, "Detection of dummy trajectories using convolutional neural networks," *Security and Communication Networks*, vol. 2019, 2019.

[9] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *ICPS '05. Proceedings. International Conference on Pervasive Services, 2005.*, 2005, pp. 88–97.

[10] X. Wu and G. Sun, "A novel dummy-based mechanism to protect privacy on trajectories," in *2014 IEEE International Conference on Data Mining Workshop*, 2014, pp. 1120–1125.

[11] Y. Li, J. Luo, C.-Y. Chow, K.-L. Chan, Y. Ding, and F. Zhang, "Growing the charging station network for electric vehicles with trajectory data analytics," in *2015 IEEE 31st International Conference on Data Engineering*, 2015, pp. 1376–1387.

[12] Y. Li, M. Steiner, J. Bao, L. Wang, and T. Zhu, "Region sampling and estimation of geosocial data with dynamic range calibration," in *2014 IEEE 30th International Conference on Data Engineering*. IEEE, 2014, pp. 1096–1107.

[13] M. Pan, Y. Li, X. Zhou, Z. Liu, R. Song, H. Lu, and J. Luo, "Dissecting the learning curve of taxi drivers: A data-driven approach," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 783–791.

[14] Y. Zheng and X. Zhou, *Computing with spatial trajectories*. Springer Science & Business Media, 2011.

[15] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 791–800.

[16] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, "Text matching as image recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

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

[21] S. Qian, B. Cheng, J. Cao, G. Xue, Y. Zhu, J. Yu, M. Li, and T. Zhang, "Detecting taxi trajectory anomaly based on spatio-temporal relations," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021.

[22] D. Chen, Y. Du, S. Xu, Y.-E. Sun, H. Huang, and G. Gao, "Online anomalous taxi trajectory detection based on multidimensional criteria," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[23] Z. Tu, K. Zhao, F. Xu, Y. Li, L. Su, and D. Jin, "Beyond k-anonymity: protect your trajectory from semantic attack," in *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2017, pp. 1–9.

[24] J. Kang, D. Steiert, D. Lin, and Y. Fu, "Movewithme: Location privacy preservation for smartphone users," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 711–724, 2019.

[25] T. Cunningham, G. Cormode, H. Ferhatosmanoglu, and D. Srivastava, "Real-world trajectory sharing with local differential privacy," *arXiv preprint arXiv:2108.02084*, 2021.

[26] J. Tang, H. Zhu, R. Lu, X. Lin, H. Li, and F. Wang, "Dlp: Achieve customizable location privacy with deceptive dummy techniques in lbs applications," *IEEE Internet of Things Journal*, vol. 9, no. 9, pp. 6969–6984, 2022.

[27] S. Dabiri and K. Heaslip, "Inferring transportation modes from gps trajectories using a convolutional neural network," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 360–371, 2018.

[28] S. Ezzini, I. Berrada, and M. Ghogho, "Who is behind the wheel? driver identification and fingerprinting," *Journal of Big Data*, vol. 5, no. 1, pp. 1–15, 2018.

[29] S. Dabiri, N. Marković, K. Heaslip, and C. K. Reddy, "A deep convolutional neural network based approach for vehicle classification using large-scale gps trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 116, p. 102644, 2020.