Security and Safety, Vol. 4, 2024018 (2025)
https://doi.org/10.1051/sands/2024018
sands.edpsciences.org

Security and Safety of Data in Cloud Computing

Research Article

OPEN ACCESS

Software Engineering

Trajectory-user linking via supervised encoding

Chengrui Hu[1], Zheng Li[2], Siyuan Wu[1], Bowen Shu[1], Min Zhang[1], and Hao Li[1,*]

[1] *Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*
[2] *CISPA Helmholtz Center for Information Security, Saarbrücken 66123, Germany*

Received: 23 July 2024 / Revised: 21 October 2024 / Accepted: 24 October 2024 / Published online: 30 January 2025

Abstract With the explosive growth of Location-Based Services (LBS), a substantial amount of geolocation data, containing end-user private information, is amassed, posing severe privacy risks. Trajectory-User Linking (TUL) is a trajectory mining task aimed at linking trajectories to their generators. Recent research has introduced deep learning-based TUL models. However, these models face challenges related to limited data quality and inadequate extraction of bidirectional and multi-topic semantic information from trajectories. In this study, we propose Trajectory-User Linking via Supervised Encoding (TULSE), centered on supervised encoding of location points and trajectories to address the TUL task. Specifically, TULSE extracts spatial and temporal information from location points through a novel method named Supervised Spatiotemporal Encoding. Additionally, TULSE employs a BiLSTM with multi-head attention to capture bidirectional and multi-topic semantics from trajectories. Furthermore, recognizing the limitations of current evaluation metrics, we introduce a novel metric named Hierarchical Privacy Loss (HPL). HPL offers a more detailed assessment of TUL solutions by statistically analyzing the distribution of prediction accuracy among users. We conduct extensive experiments on two benchmark datasets, and empirical results show that TULSE outperforms existing TUL methods.

Keywords Trajectory-user linking, Deep learning, Location embedding, Attention mechanism, Privacy risk measurement

Citation Hu C, Li Z and Wu S et al. Trajectory-user linking via supervised encoding. Security and Safety 2025; **4**: 2024018. https://doi.org/10.1051/sands/2024018

1 Introduction

In recent years, the rapid expansion of Location-Based Services (LBS) has enabled service providers to amass substantial trajectory data expeditiously. This data holds the potential to comprehend and forecast human movement patterns, facilitating personalized business development and enhancing user experience. Nevertheless, this information encompasses sensitive user privacy details, encompassing geographical location and daily life trajectory. As a result, many users are reluctant to disclose personal information when providing trajectory data, and most LBS providers actively anonymize trajectory data before disseminating it to avoid privacy leakage.

The Trajectory-User Linking (TUL) algorithm, as introduced by Gao *et al.* [1], is a recently developed trajectory mining approach that establishes links between trajectories and their originators through the examination of user and trajectory movement patterns. While TUL methodologies hold the potential to advance the creation of personalized services, it is crucial to acknowledge that an effective TUL scheme

* Corresponding author (email: lihao@iscas.ac.cn)

may concurrently pose a significant risk of privacy leakage. For instance, an attacker could ascertain users' health statuses by discerning the generators of trajectories that have consistently lingered within hospital premises. Consequently, delving into research on the TUL problem proves valuable not only for enhancing Location-Based Services (LBS) but also scrutinizing the inherent privacy risks associated with trajectory datasets.

To tackle the TUL challenge, prior strategies for the TUL task relied on methodologies grounded in Longest Common Subsequence (LCSS) [2], Dynamic Time Warping (DTW) [3], or comparable algorithms. These methodologies computed the similarity between known and unknown users' trajectories. However, they fell short in extracting semantic information from trajectories and proved sensitive to trajectory length and noise. In more recent developments, researchers have endeavored to model trajectory features using probabilistic models capable of discerning dependencies among location points. A notable approach involves leveraging the Markov hypothesis, wherein an estimated transition matrix is employed to forecast a user's future actions. Nonetheless, trajectories may not universally conform to Markov properties, posing challenges for Markov-based models in capturing long-term dependencies. Consequently, these solutions exhibit limited efficacy in addressing the TUL problem.

In recent years, the swift progress in deep learning has led to the integration of deep learning models into TUL approaches. Recognizing the inherent serial nature of both trajectory data and natural language, many solutions, drawing inspiration from current advancements in Natural Language Processing (NLP), typically treat trajectories as sentences and view location points as words. Specifically, concerning location points, prevalent approaches commonly employ the word2vec technique [4] to embed each point into a vector, transforming a trajectory into a sequence of vectors. Among the methods dedicated to analyzing trajectory semantics, the predominant strategies often incorporate Recurrent Neural Network (RNN)-based models. Notably, Gao *et al.* [1] were among the pioneers in utilizing Long Short Term Memory (LSTM) and Gate Recurrent Unit (GRU) for trajectory analysis. Additionally, some solutions integrate the attention mechanism into trajectory analysis, such as Sun *et al.* [5] employing attention-based RNN, and Li *et al.* [6] utilizing a multi-head attention mechanism to capture the semantic information of trajectories. Despite their accomplishments in TUL tasks, these solutions still grapple with ongoing challenges.

1) **Data with poor quality and small scale.** Datasets within the TUL domain exhibit considerably smaller scales compared to the corpora in the NLP field. Notably, certain TUL datasets, such as check-in datasets, present markedly shorter trajectory lengths in comparison to those observed in natural language datasets. The constrained scale and brevity of trajectories, particularly in the case of check-in datasets, pose formidable challenges in fully leveraging the spatial distribution and temporal information of location points within the word2vec framework. Consequently, extant methods for embedding location points based on word2vec, as evidenced in works such as those by Gao *et al.* [1], Sun *et al.* [5], Wang *et al.* [7], Yu *et al.* [8], Miao *et al.* [9], Chen *et al.* [10], encounter difficulties in acquiring high-quality representations of location points in the current TUL datasets. This limitation significantly impacts the efficacy of TUL approaches.
2) **Insufficient understanding of bidirectional, multi-topic semantic information.** Trajectories often manifest not only forward seriality but also backward seriality, indicating that location points can be influenced by future trajectory plans. However, the majority of existing solutions predominantly focus on forward seriality. Furthermore, trajectories may encapsulate intricate semantic information, especially in scenarios where individuals undertake multi-purpose travel, such as going out for dinner before engaging in shopping activities. Consequently, prevailing neural network models utilized in TUL research exhibit limitations in comprehending the nuanced semantics inherent in trajectories.
3) **Inadequate privacy risk assessment.** Existing evaluation metrics, namely ACC@K and macro-F1, operate on an average scale, lacking nuanced measurements. ACC@K gauges the accuracy of predicting labels ranked within the top k by prediction probabilities, while macro-F1 represents the harmonic mean of precision and recall in classification. Nevertheless, there is a notable gap in addressing the gravity of a reliable attack aimed at a small subset of samples within the entire dataset. To illustrate, consider two datasets, each with users possessing an equal number of trajectories. In one dataset, the TUL approach attains 90% and 10% accuracy in top-1 prediction probabilities for two users, while in the other, it achieves 51% and 49%. Although both datasets yield the same 50% ACC@1 score, it is

evident that the privacy risk for certain users is underestimated. Consequently, there is a compelling need for more fine-grained metrics to offer a precise assessment of the mentioned risks.

To tackle the challenges outlined earlier, we introduce TULSE, a solution grounded in the supervised encoding of both location points and trajectories. In addressing the initial challenge, we propose Supervised Spatiotemporal Encoding. This approach employs statistical methods to encode spatial and temporal features of location points, facilitating the representation vectors to express semantic information pertaining to user preferences and location categories. To confront the second challenge, we devise a bidirectional sequential encoder that captures more comprehensive trajectory features. This encoder leverages Bidirectional Long Short-Term Memory (BiLSTM) to acquire bidirectional sequential semantics and integrates an attention mechanism to capture implicit multi-topic semantics within trajectories. Additionally, we introduce a novel metric termed Hierarchical Privacy Loss (HPL). HPL performs statistical analysis on prediction accuracy across different users and constructs a distribution of user privacy loss attributed to the TUL solution. Therefore HPL performs a detailed analysis of privacy risk by evaluating prediction accuracy across different users and reveals privacy risk disparities that are neglected in conventional TUL metrics.

In summary, our main contributions are as follows:

1) We introduce a novel location point encoding method called Supervised Spatiotemporal Encoding to incorporate spatial and temporal features into location point representation vectors. This method utilizes two types of vectors, named Access Feature and Dayparting Frequency Vector. The Access Feature is built based on the spatial frequency of user visits to locations and can represent user preference information. We generate the Dayparting Frequency Vector by analyzing the temporal frequency of location visits, which can express the category feature. These two vectors are combined through concatenation to be spatiotemporal representation vectors of location points. This approach significantly improves the quality of the representation vectors, enabling more effective semantic features related to user preferences and location categories.
2) We design an attention-based BiLSTM method. It employs BiLSTM to capture bidirectional trajectory sequential information and utilizes a multi-head attention mechanism to map trajectories into different representation subspaces. The attention mechanism enables extracting various mobility pattern features, namely multi-topic semantic information, in trajectories. We have also explored transformers capable of capturing sequential patterns, demonstrating the effectiveness of the attention-based BiLSTM approach in identifying the trajectory generators.
3) We introduce a novel metric called Hierarchical Privacy Loss (HPL), which provides a more fine-grained measurement of the privacy loss revealed by the TUL solution by statistically analyzing the distribution of prediction accuracy among different users. For example, even though TULSE achieves only a 68% accuracy on the Gowalla-D dataset, it attains a 90% accuracy for 20% of the users. This indicates that a subset of users experiences significantly higher privacy leakage, a critical aspect that traditional metrics fail to capture or express effectively.
4) We conduct experiments on both check-in and GPS datasets, which have significantly different sampling rates. The experiments demonstrate that our solution can outperform all state-of-the-art TUL approaches with acceptable spatial and temporal costs.

The rest of the paper is organized as follows: Section 2 introduces the notations and formulation of the TUL problem and discusses related work. Section 3 details our novel model TULSE and the novel metric HPL. In Sections 4, 5, and 6, we evaluate TULSE to show the effectiveness of its components and also analyze results from HPL. Finally, we conclude the paper in Section 7.

2 Background and related work

In this section, we first introduce the notations and problem formulation of the TUL problem. Then, we will provide a brief overview of the main related work of the TUL task.

2.1 Formulation

The trajectory data is collected with millions of location points in a region. Each location point is composed of spatial and temporal contexts.

Definition 1 (Location Point). A location point $p_i = (l_i, t_i)$ consists of the coordinate of the location point $l_i = (longitude_i, latitude_i)$ and the timestamp t_i when user generates the it.
Definition 2 (Trajectory). A trajectory $T = \{p_1, p_2, \cdots, p_m$ represents a trajectory that contains m location points sorted by time order.
Problem 1 (Trajectory-User Linking). The solution to the TUL problem aims to find a mapping relationship that maps anonymous trajectories to their generators. Formally, let $\mathbf{T} = \{T_1, T_2, \cdots, T_k\}$ be the trajectory dataset with k trajectories, and $\mathbf{U} = \{u_1, u_2, \cdots, u_n\}$ be the set of all n users u_i, then the solution to TUL problem is to find a function $f(T_i) = u_j$.

2.2 Related work

Early methods based on explicit features [2, 3, 11–15] typically involved the extraction of explicit features from trajectories, such as trajectory length, trajectory point location information, time intervals within trajectories, and then calculating distances or similarities between trajectories based on the extracted features. Some methods computed trajectory similarities based on the top K trajectory points in terms of visit frequency [11], interest points [12], or random points [13], while others used LCSS [2], DTW [3], or edit distance on real sequence (EDR) [14] algorithms to calculate trajectory similarities. These methods have the advantage of intuitively measuring similarity between trajectories but often neglect the exploration of potential user mobility patterns. They are also sensitive to both trajectory length and noise. As a result, they are not suitable for practical applications.

Due to the seriality of trajectory data, some researchers [16–19] choose to model and analyze user trajectory data using probabilistic models. These models are generally based on the assumption that trajectories adhere to the Markov property, which, in the context of the TUL problem, implies that the location of a point depends only on all previous points visited before that point. Researchers employed Markov models [16], Hidden Markov Models [17], Bayesian Models [18], and Dynamic Bayesian Models [19] to model user trajectories, improving the performance of trajectory data mining. The probability models fitted to the data also partially reflect the spatial and temporal activity preferences of users [20]. However, the assumption that trajectories strictly adhere to Markov properties does not always hold. On one hand, a location point is generally influenced by future location points, indicating that users plan their future trajectories in advance. On the other hand, some datasets, such as check-in datasets, exhibit sparse temporal and spatial distributions, resulting in weak correlations between different location points. Furthermore, these models have limitations when it comes to analyzing long-term dependencies of location points.

In recent years, inspired by the rapid developments in the NLP field and other trajectory analysis areas, many TUL solutions that were based on location embedding and RNN were proposed [1, 7, 8, 21]. Gao *et al.* [1] introduced the first formalization of the TUL problem and designed a TUL solution based on the semantic extraction of location points and RNN. Subsequently, Zhou *et al.* [21] introduced Variational Autoencoders (VAE) to extract more complex hidden features in trajectories. Wang *et al.* [7] used a dual-objective neural network to simultaneously learn the user's mobility pattern features and trajectory semantic features. Apart from considering TUL as a classification task, some research focuses on indirectly improving TUL accuracy by focusing on high-quality representations of trajectory users. For example, Yu *et al.* [8] designed Siamese Neural Networks to learn trajectory vectors and used K-Nearest Neighbors (KNN) [22] to solve the TUL problem. However, these solutions primarily focus on extracting sequential features from trajectories while overlooking the semantics of individual location points within the trajectories and the contributions of different location points to information extraction. Therefore, it is challenging for them to comprehend the mobility pattern features of trajectories fully.

Recently, inspired by the widespread application of attention mechanisms in the NLP field, attention mechanisms are also used to address the TUL problem [5, 9, 10]. Sun *et al.* [5] employ a single-headed self-attention mechanism to calculate attention scores, representing the importance of different location points within trajectories. Miao *et al.* [9] generate representation vectors for historical trajectories using a historical attention model, thereby extracting multi-periodic mobility patterns in trajectories. Chen *et al.* [10] use mutual distillation learning to train two models, an RNN and a Transformer. The RNN captures the mobility pattern features of trajectories, while the Transformer captures longer-term dependencies in enhanced trajectories. However, existing solutions often lack attention to the bidirectional

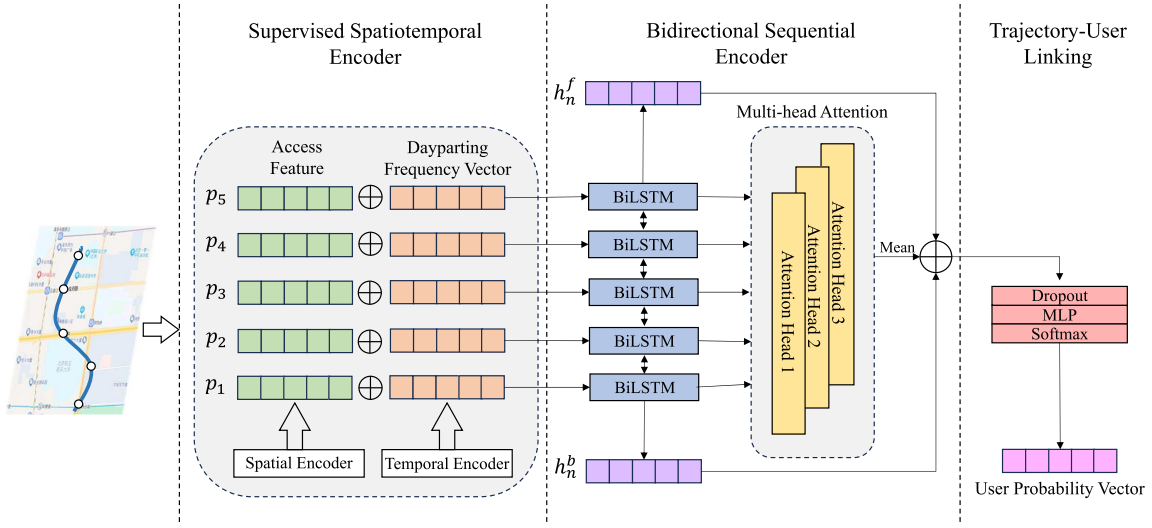

Figure 1. The architecture of TULSE model

sequence relationships and various potential mobility patterns commonly found in human movement trajectories, which are crucial for addressing the TUL problem.

Additionally, many researchers argue that the embedding scheme for location points is important for improving the TUL solution. Inspired by the NLP field, most researchers [1, 5, 7, 23] adopt location point embedding based on word2vec. Gao *et al.* [1] and Sun *et al.* [5] directly apply the word2vec method to embed trajectories. Wang *et al.* [7] additionally combine the vector generated by word2vec with the time information when the user visits it and the category information of that point to represent location points. However, due to the sparsity of location points and the relatively small size of the corpus in the TUL field, such word2vec-based schemes and their variations fail to fully capture the spatiotemporal semantics of location points and therefore affecting the performance of TUL tasks. To address this, Hu *et al.* [23] and Wang *et al.* [7] use random walk algorithms, respectively, on weighted and unweighted graphs to augment the corpus, improving the effectiveness of location point embedding. However, since the generated graphs do not fully leverage the temporal distribution characteristics of trajectories, some of the trajectories produced by random walks are unreasonable, resulting in limited improvements compared to previous solutions. Li *et al.* [6] propose an approximate one-hot encoding method to avoid semantic loss caused by insufficient corpus, but it lacks the capture of spatial and temporal information of location points.

3 TULSE

This section begins with an overview of the method, followed by the introduction of each component individually. Additionally, to provide a more detailed analysis of the privacy risks unveiled by TUL solutions, we introduce a novel metric termed HPL.

3.1 Overview

The comprehensive framework of TULSE is depicted in Figure 1 and is comprised of three primary components: a Supervised Spatiotemporal Encoder, a Bidirectional Sequential Encoder, and a Trajectory-User Linking model. The Supervised Spatiotemporal Encoder utilizes statistical methods to encode spatial and temporal characteristics of location points into a low-dimensional vector. Subsequently, the representation vectors, corresponding to the sequence of location points within a trajectory, are processed by the Bidirectional Sequential Encoder to capture both sequential and contextual semantic information. Finally, the Multi-Layer Perceptron (MLP), a widely employed classifier, efficiently establishes a connection between the extracted trajectory information and its generator.

3.2 Data preprocessing

First, we partition the spatial region where location points are situated into several non-overlapping grids, denoted as $g_i = (x_i, y_i)$, where (x_i, y_i) represents the coordinate of the grid. Then, for each location point $p_i = (l_i, t_i)$, we replace its coordinate l_i with the coordinate of the grid in which it is located, namely $p'_i = (g_i, t_i)$.

Next, following previous works [1, 5, 6], for a single user, we initially connect all the location points in chronological order to form a trajectory. Subsequently, we divide this trajectory into several sub-trajectories where adjacent nodes with a time difference exceeding 6 hours are assigned to two separate sub-trajectories.

Trajectories, characterized by a high sampling rate, often become excessively long, which leads to an abundance of location points clustered within a grid. This poses a challenge for RNNs to effectively extract meaningful trajectory features. In order to address this issue and retain the original semantics of the trajectory, we have implemented a technique that involves sampling points from a trajectory. These sampled points adhere to the following formula.

$$t_{i+k} - t_i \geq a^k, \text{if } \forall 0 < j \leq k, g_{i+j} = g_i \tag{1}$$

where a is a hyperparameter. This approach can compress the number of points within a grid to $O(\log \Delta t)$, where Δt represents the maximum time difference between consecutive location points within a grid. This method can ensure that the number of location points remains positively correlated with Δt while reducing the number of location points in one grid to an acceptable range.

3.3 Supervised spatiotemporal encoder

For the location points p'_i, we introduce a new representation method named Supervised Spatiotemporal Encoding. This method computes two vectors for every location point: the Access Feature and the Dayparting Frequency Vector.

1) **Access Feature.** Yves-Alexandre *et al.* [13] reveal that, among mobile phone users, randomly sampling four points on a trajectory has a 95% probability of identifying its generator. Therefore, trajectories generated by different users typically exhibit significant spatial uniqueness. Inspired by this, we leverage user access patterns to location points as Access Feature vectors for them, which can express user preference. Specifically, for each location point, we create an Access Feature vector $v^{af} \in \mathbf{R}^n$, where n represents the number of users, and the i-th dimension of it corresponds to the frequency of user u_i accessing that location point. Subsequently, we normalize the vector v^{af} to reduce the sensitivity of our approach to feature scales.
2) **Dayparting Frequency Vector.** Given that visits to certain locations at different times may convey distinct meanings, such as a visit to a large supermarket at 10 AM indicating shopping, whereas a visit at noon may imply having a meal, the temporal distribution of user visits to location points harbors valuable semantic information regarding their categories. To capture this nuanced information, we analyze the temporal features of user visits, creating Dayparting Frequency Vectors for each point to reflect its category characteristics.

While some datasets provide category information for location points, numerous datasets, such as GPS datasets, lack this information. Dayparting Frequency Vectors offer a solution to conflicts arising when the same grid encompasses different categories. Specifically, we divide a day into 24 time slots by the hour, creating a Dayparting Frequency Vector $v^{df} \in \mathbb{R}^{24}$ for each location point. The i-th dimension represents the frequency of visits during the i-th time slot. To bolster the model's robustness while preserving relative time semantics across location points, we standardize vectors comprising dimensions corresponding to each time slot for all location points.

We concatenate v^{af} and v^{df} as the spatiotemporal representation vector, namely $v_i = v_i^{af} \| v_i^{df}$. Finally, we join all the representation vectors of location points in a sequence as a sequence to represent it.

3.4 Bidirectional sequential encoder

In reality, an individual's trajectory is usually influenced not only by their previous trajectories but also by their future trajectories, which represent their plans for upcoming actions. Furthermore, a single trajectory may have multiple purposes. To capture the bidirectional and multi-topic semantics of trajectories, we input the vector sequences of the trajectory generated in subsection 3.3 into a bidirectional sequential encoder. This encoder comprises two components: a BiLSTM model and an attention mechanism.

Our architecture of the BiLSTM model includes a forward LSTM and a backward LSTM. A unidirectional LSTM consists of hidden state h, input gate i, output gate o, memory cell c, and output h. For an encoded trajectory $T = \{v_1, v_2, ..., v_m\}$ with m location points, the t-th output calculation formula for a single LSTM is as follows:

$$i_t = \sigma(W_i v_t + U_i v_{t-1} + b_i) \tag{2}$$
$$f_t = \sigma(W_f v_t + U_f v_{t-1} + b_f) \tag{3}$$
$$o_t = \sigma(W_f v_t + U_o v_{t-1} + b_o) \tag{4}$$
$$g_t = \tanh(W_g v_t + U_g h_{t-1} + b_g) \tag{5}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot g_t \tag{6}$$
$$h_t = o_t \cdot \tanh(c_t) \tag{7}$$

where $\sigma(\cdot)$ and $\tanh(\cdot)$ refer to the sigmoid and hyperbolic tangent activation functions, respectively. Regarding the BiLSTM, the t-th output of forward and backward LSTM are denoted as h_t^f and h_t^b, respectively, and the t-th output of BiLSTM is the concatenation of h_t^f and h_{m-t}^b, *i.e.*, $h_t = h_t^f || h_{m-t}^b$. We use H to represent the horizontal concatenation of all the output of BiLSTM, and $h_{out} = h_m^f || h_m^b$ as the combination of two final outputs of BiLSTM.

Given the complexity of semantics within trajectories, we have implemented a multi-head attention mechanism to better capture and process these intricate patterns. This mechanism allows for simultaneous attention to information from various representation subspaces at different positions, thus capturing the multi-topic complex motion features within trajectories. The multi-head attention mechanism used in this paper is outlined as follows:

$$H_i' = H * W_i \tag{8}$$
$$\alpha_{ij} = softmax(h_j^T H_i') \tag{9}$$
$$head_{ij} = H_i' \alpha_{ij}^T \tag{10}$$
$$context_j = head_{0j} \| head_{1j} \| \cdots \| head_{kj} \tag{11}$$
$$v_{att} = Mean(context_0, context_1, \cdots, context_m) \tag{12}$$

where α_{ij} refers to the attention score in i-th attention head for j-th output of BiLSTM; $head_{ij}$ refers to the context in i-th subspace for the h_j; $context_j$ represents the context of j-th input of BiLSTM; v_{att} represents the context vector for the whole trajectory; W_i is learnable parameter matrix; $*$ is element-wise product; and $Mean(\cdot)$ is an element-wise mean function.

Finally, we concatenate v_{att} with h_{out} as the semantic vector of the trajectory.

Apart from the attention-based BiLSTM model, we also explored the transformer model, which has a significant impact in the field of NLP for effectively capturing semantics. However, we found that its performance was not as good as the attention-based BiLSTM. A detailed experimental analysis refers to subsection 5.4.

3.5 Trajectory-user linking module and hierarchical privacy loss

The purpose of the trajectory-user linking model is to calculate the likelihood of the trajectory semantic vectors obtained in the previous steps belonging to each user. As shown in Figure 1, we employ an MLP to construct this module and use the softmax function to obtain the final user probability vector. We also utilize the Dropout method to prevent overfitting of the model.

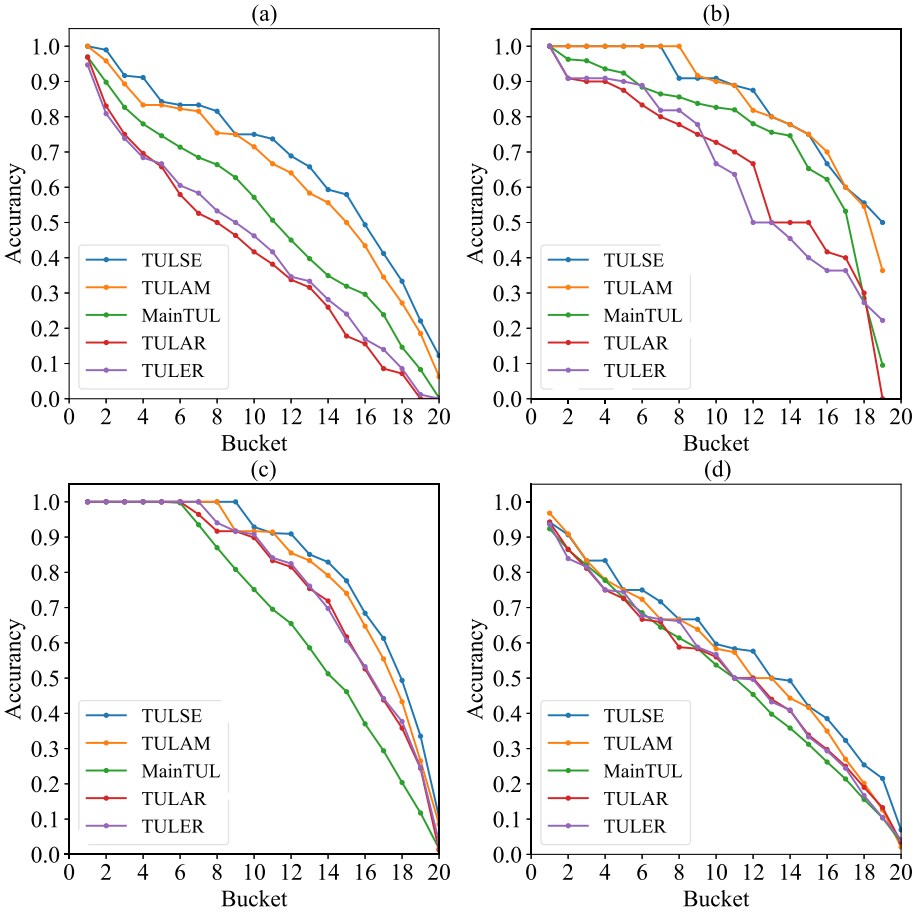

Figure 2. Comparison of HPL metric on four datasets: (a) Gowalla-D; (b) Gowalla-S; (c) Brightkite; (d) Foursquare

As the TUL problem can be considered as a multi-class classification problem, we use cross-entropy as the loss function, which is defined as follows:

$$Loss = -\frac{1}{k}\sum_{i=1}^{k} d_i \log(p_i) \tag{13}$$

where k represents the total number of trajectories in the training set, d_i is the one-hot representation vector of the ground truth for the i-th trajectory, and p_i is the predicted user probability vector for the i-th trajectory by the model.

To provide a more detailed analysis of the effectiveness of certain approaches, we introduce a new metric, Hierarchical Privacy Loss (HPL). HPL begins by calculating the top-1 prediction accuracy for each user, followed by ranking the accuracies across all users. The ranked accuracies are then divided into several non-overlapping, equally sized subgroups, referred to as buckets. The average accuracy for each bucket is subsequently computed, generating a set of prediction accuracy distributions, as depicted in Figure 2. In this graph, the horizontal axis represents the bucket index, the vertical axis represents accuracy, and the coordinates of each point represent the average accuracy for the corresponding bucket. HPL enables the assessment of how a TUL approach distributes privacy leakage across different users, providing enhanced interpretability and a fine-grained measurement of privacy loss within the dataset.

Table 1. Statistics of datasets

Dataset	Users	Trajectories	Location points	Average length
Gowalla-D	143	20335	62691	3.08
Gowalla-S	19	3587	11040	3.08
Foursquare	242	38097	117995	3.10
Brightkite	141	41934	85777	2.05
Geolife	78	12398	1624460	131.03

Table 2. Number of trajectories in total, training and testing datasets for all the datasets

Dataset	Total trajectories	Training trajectories	Testing trajectories
Gowalla-D	20335	18619	1716
Gowalla-S	3587	3359	228
Foursquare	38097	35193	2904
Brightkite	41934	40242	1692
Geolife	12398	11462	936

4 Experimental setup

4.1 Datasets

To comprehensively demonstrate the generality of our proposed solution, we conduct experiments on two types of real-world datasets: check-in datasets and GPS datasets.

(1) **Check-in dataset.** The check-in datasets record users' check-in activities in LBS networks. The location points in this dataset are relatively sparse, and the trajectory lengths are much shorter. The check-in datasets we use include three datasets: Gowalla [24], Foursquare [20], and Brightkite [24]. Considering the uneven distribution of location points in these datasets, we selected a dense area and a sparse area from the Gowalla dataset, referred to as Gowalla-D and Gowalla-S, respectively, to showcase the generality of our solution across datasets with varying densities. For the other two datasets, we only selected a dense area for experimentation.
(2) **GPS dataset.** The GPS datasets record users' GPS data in LBS networks. The distribution of location points in this dataset is dense, and the trajectory lengths are significantly longer than those in the check-in datasets. The GPS dataset we used is the Geolife [25–27] dataset.

Table 1 shows the number of users, trajectories, and location points, and the average length of trajectories for each dataset.

In the case of dataset partitioning, we ensure that the test dataset is balanced, meaning that each user has an equal number of trajectories in the test dataset to ensure consistent user contributions to the experimental results. Table 2 shows how we partition each dataset.

4.2 Baselines

To validate the effectiveness of TULSE, we compared it with the following TUL methods as baselines:

(1) **LCSS.** The Longest Common Subsequence method (LCSS) computes the length of the Longest Common Subsequence between two trajectories as their similarity measure. In the test dataset, each trajectory is assigned the user label from the most similar trajectory in the training dataset.
(2) **LDA.** linear discriminant analysis (LDA) is a classical spatial data classification model. We embedded trajectories using Term Frequency-Inverse Document Frequency (TF-IDF) [4] and then used LDA to classify the embedded vectors.
(3) **SVM.** Support Vector Machine (SVM) is a powerful machine learning algorithm capable of handling high-dimensional data and nonlinear relationships. Similar to LDA, we used TF-IDF to embed trajectories and then employed SVM for classification.

(4) **TULER.** TULER [1] encodes location points using word2vec and uses an RNN to extract temporal information from trajectories to link them to their generators.
(5) **TULAR.** TULAR [5] employs a neural network model based on an attention mechanism that focuses on selected parts of source trajectories.
(6) **MainTUL.** MainTUL [10] employs a distillation learning method to train both RNN and transformer models. The former captures movement pattern features in trajectories, while the latter captures longer-term dependencies. It is only tested on check-in datasets because it is not suitable for high sampling rate data.
(7) **TULAM.** TULAM [6] introduces a novel approximate one-hot location point embedding scheme and uses an RNN with a multi-head attention mechanism to capture trajectory semantics.

Among the baseline methods considered, LCSS, LDA, and SVM do not incorporate learning of sequential information. Additionally, we include TULER and TULAR as representatives of traditional deep neural network approaches. To showcase the efficacy of TULSE, we conduct a comparative analysis with two state-of-the-art methods, MainTUL and TULAM.

4.3 Evaluation metrics

We utilized three primary metrics: ACC@K, macro-F1, and Hierarchical Privacy Loss (HPL). In the case of ACC@K, we sorted the predicted user probability values for trajectory T_i and assessed whether the top K-ranked user labels encompass the generator of T_i. Macro-F1, representing the harmonic mean of precision and recall, is calculated using the following formula:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{14}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{15}$$

$$macro\text{-}P = \frac{1}{n}\sum_{i=1}^{n} Precision_i \tag{16}$$

$$macro\text{-}R = \frac{1}{n}\sum_{i=1}^{n} Recall_i \tag{17}$$

$$macro\text{-}F1 = \frac{2 \times macro\text{-}P \times macro\text{-}R}{macro\text{-}P + macro\text{-}R} \tag{18}$$

where TP_i, FP_i, and FN_i refer to the rate of the true positive, false positive, and false negative of user u_i, respectively, and n is the number of users.

HPL is a newly introduced metric in this paper, and its specific calculation method can be found in subsection 3.5. We conducted this more detailed analysis on our method and some baselines, including the two state-of-the-art approaches, MainTUL and TULAM. The experimental result can show the effectiveness of TULSE and the importance of HPL.

4.4 Parameter settings

The experimental parameters are detailed in Table 3. To account for deviation and noise in location points, we select a grid size ranging from 100 to 500 meters. The dimensions of hidden states in BiLSTM are set to 400. In the Trajectory-User Linking model, we employ an MLP with 2 layers. Based on experimental performance, our method utilizes attention with 6 heads. The model is trained using the Adam algorithm [28], with Dropout [29] implemented to prevent overfitting, and the dropout rate is set to 0.6. The initial learning rate is established at 0.001, gradually decreasing by 0.8 every 10 epochs until it reaches no less than 0.0001.

5 Experimental results

In this section, we evaluate TULSE in subsection 5.1, and analyze the result obtained from HPL in subsection 5.2. Then, subsection 5.3 and subsection 5.4 explore the impact of encoding methods for

Table 3. Parameters used in experiment

Parameters	Value
Grid size	100–500 meters
Hidden size	400
Attention heads	6
MLP layers	2
Learning rate	0.001−0.0001, decays by 0.8 every 10 epochs
Dropout rate	0.6

Table 4. Experimental results on Gowalla-D dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LCSS	0.4149	0.4114	0.6026	0.6766	0.7535
LDA	0.5934	0.5787	0.6888	0.7255	0.7611
SVM	0.6378	0.6200	0.7523	0.8065	0.8689
TULER	0.4400	0.4387	0.5655	0.6166	0.6845
TULAR	0.4494	0.4213	0.5451	0.6046	0.6785
MainTUL	0.5371	0.5822	0.6687	0.6910	0.7222
TULAM	0.6486	0.6376	0.7674	0.8089	0.8528
TULSE	**0.6889**	**0.6815**	**0.8041**	**0.8438**	**0.8882**

Table 5. Experimental results on Gowalla-S dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LCSS	0.7078	0.6667	0.8904	0.9430	0.9781
LDA	0.8034	0.7851	0.8947	0.9254	0.9474
SVM	0.8289	0.8202	0.9561	0.9781	0.9868
TULER	0.6510	0.6458	0.8177	0.8698	0.9531
TULAR	0.6704	0.6458	0.8073	0.8958	0.9531
MainTUL	0.7539	0.7931	0.9200	0.9476	0.9766
TULAM	0.8456	0.8385	0.9375	0.9635	0.9896
TULSE	**0.8556**	**0.8490**	**0.9688**	**0.9948**	**0.9948**

location points and trajectories. Finally, since the number of attention heads can significantly affect the result and complexity of our model, we conduct an experiment about it in subsection 5.5.

5.1 Performance on TUL task

The experimental results are presented in Tables 4, 5, 6, 7, and 8, with the best-performing outcomes highlighted in **bold** and the second-best results underlined. Our proposed TULSE approach demonstrated superior performance on both check-in and GPS datasets. For instance, as illustrated in Table 4, our results for the Gowalla-D dataset include 68.15% and 68.89% in ACC@1 and macro-F1, respectively. In comparison, TULAM achieved 63.76% and 64.86%, while MainTUL attained only 53.71% and 58.22% in the same metrics. This superiority can be attributed to two key factors. Firstly, TULSE leverages the Supervised Spatiotemporal Encoder, enhancing the utilization of spatial and temporal information from location points. Secondly, using BiLSTM with multi-head attention accounts for both bidirectional and multi-topic semantic information, contributing to superior performance.

Furthermore, as evident from Tables 4, 5, 6 and 7, in check-in datasets characterized by poor data quality, despite the application of RNN and attention mechanisms, the state-of-the-art method MainTUL occasionally exhibits inferior performance compared to Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA), which do not analyze sequential information in trajectories. This can be attributed to MainTUL's inadequate extraction of spatial and temporal information and its neglect

Table 6. Experimental results on Foursquare dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LCSS	0.0836	0.0668	0.1477	0.2042	0.3130
LDA	0.4103	0.3843	0.5444	0.6129	0.7008
SVM	0.4898	0.4728	0.6291	0.6890	0.7638
TULER	0.5330	0.5156	0.6538	0.7052	0.7705
TULAR	0.5340	0.5160	0.6497	0.6986	0.7708
MainTUL	0.5124	0.5438	0.6593	0.7016	0.7445
TULAM	0.5658	0.5493	0.6813	0.7378	0.8017
TULSE	**0.5976**	**0.5771**	**0.7028**	**0.7476**	**0.8146**

Table 7. Experimental results on Brightkite dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LCSS	0.5201	0.5538	0.7210	0.8014	0.8717
LDA	0.7522	0.7429	0.8528	0.8783	0.9001
SVM	0.7983	0.7908	0.8978	0.9255	0.9504
TULER	0.7661	0.7675	0.8700	0.8969	0.9225
TULAR	0.7665	0.7613	0.8613	0.8956	0.9256
MainTUL	0.6708	0.6929	0.7745	0.7993	0.8250
TULAM	0.8063	0.7987	0.9014	0.9303	0.9525
TULSE	**0.8344**	**0.8294**	**0.9294**	**0.9450**	**0.9694**

Table 8. Experimental results on Geolife dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LCSS	0.3147	0.2265	0.4583	0.5684	0.7447
LDA	0.5343	0.5182	0.6912	0.7318	0.7853
SVM	0.6281	0.5972	0.7703	0.8365	0.9017
TULER	0.6791	0.6585	0.8359	0.8783	0.9230
TULAR	0.6714	0.6585	0.8393	**0.8962**	**0.9308**
TULAM	0.6545	0.6350	0.8270	0.8661	0.9040
TULSE	**0.6887**	**0.6674**	**0.8415**	0.8940	0.9263

of bidirectional semantics. Consequently, these findings underscore the significance of the previously mentioned information in the TUL solution. Approaches that disregard such considerations may yield performance even worse than traditional methods.

Due to the higher sampling rate, GPS datasets exhibit larger corpora and superior data quality compared to check-in datasets. As illustrated in Table 8, all deep learning-based methods outperform SVM and LDA. This underscores the significance of seriality in the TUL field.

5.2 Evaluation with HPL

The comparison of the experimental HPL metric is presented in Figure 2. In accordance with subsection 3.5, we compute the accuracy of TUL methods for all users, arranging them in descending order. Subsequently, we categorize them into 20 buckets, with each bucket encompassing 5% of users in the dataset. This figure enables the observation that, in contrast to other commonly used average-based metrics, our approach facilitates a comprehensive perspective on user privacy loss, allowing for a more detailed analysis of the solution's efficacy.

In contrast to other metrics, each approach consistently achieves a 90% accuracy in correctly identifying the trajectories of 5% of users (1 bucket) across all datasets, signifying a high privacy risk for these users. Furthermore, in the Gowalla-D and Foursquare datasets, TULSE records ACC@1 values of

Table 9. Experimental results of different encoding methods of location points

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
Word2vec	0.3292	0.3215	0.4405	0.5066	0.6100
Approximate Onehot	0.6331	0.6196	0.7542	0.8011	0.8582
Dayparting Frequency Vector	0.5098	0.4772	0.5889	0.6340	0.7097
Access Feature	0.6778	0.6701	0.7975	0.8395	0.8792
Supervised Spatiotemporal Encoder	**0.6889**	**0.6815**	**0.8041**	**0.8438**	**0.8882**

Table 10. Experimental results of different encoding method of trajectories

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
Transformer	0.6686	0.6617	0.7933	0.8323	0.8792
Bidirectional Sequential Encoder	0.6889	0.6815	0.8041	0.8438	0.8882

68% and 57.71%, respectively. Despite this, it effectively identifies the trajectories of 20% (4 buckets) and 10% (2 buckets) of users with 90% accuracy, suggesting TULSE has the potential to cause substantial privacy leaks for these users. These nuanced insights, unique to TULSE, underscore the importance of understanding the extent of user privacy exposure, providing valuable information on the risk of privacy breaches that cannot be drawn from other metrics.

In comparison to the baselines, our results consistently demonstrate superiority across the majority of buckets, indicating that TULSE excels in performance for the majority of users. Notably, our method surpasses most solutions for users with lower privacy risks, specifically those within buckets with indices exceeding 10. This suggests the robust capability of our approach in extracting user mobility pattern features. That is, TULSE proves adept at more effectively extracting features for users with intricate mobility patterns, encompassing spatial and temporal information in location points, as well as bidirectional and multi-topic semantics in trajectories. Consequently, our approach exhibits a heightened attack capability for users previously considered relatively secure in earlier methods.

5.3 Impact of location point encoding methods

To assess the impact of various location point encoding methods on the results, we compare the Supervised Spatiotemporal Encoder employed in TULSE with two other encoding methods utilized in prior studies: approximate onehot [6] and word2vec [1, 5, 7, 23], specifically on the Gowalla-D dataset. Furthermore, as the Supervised Spatiotemporal Encoder comprises the Access Feature and the Dayparting Frequency Vector, we conducted experiments with two distinct TULSE models, each employing either the Access Feature or the Dayparting Frequency Vector. As illustrated in Table 9, the Supervised Spatiotemporal Encoder yields optimal performance by encoding the spatial and temporal features of location points into the representation vectors.

It is worth noting that the word2vec method performs the worst among all the methods, indicating that the word2vec method suffers from significant information loss on the check-in dataset.

The Access Feature performs better than the Dayparting Frequency Vector, and both two methods are relatively less effective than the whole Supervised Spatiotemporal Encoder. This suggests that the Access Feature extracts highly user-discriminative characteristics, and the Dayparting Frequency Vector provides supplementary information to improve the solution's performance.

5.4 Impact of encoding methods for trajectories

In order to more effectively capture trajectory semantics, regarding the trajectory encoder, we compare our Bidirectional Sequential Encoder with Transformer, a commonly used semantic extraction model in the NLP field that also applies the attention technique, and the results are shown in Table 10.

Table 11. Experimental results on different numbers of attention heads

Number of heads	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
1	0.5950	0.5877	0.7025	0.7500	0.8233
2	0.6131	0.6004	0.7404	0.7843	0.8558
4	0.6845	0.6737	0.7933	0.8347	0.8774
6	0.6889	0.6815	0.8041	0.8438	0.8882
10	0.6839	0.6749	0.7969	0.8407	0.8810

Table 12. Comparison of AAF, LDA, and SVM on Foursquare dataset

Method	Macro-F1	ACC@1	ACC@3	ACC@5	ACC@10
LDA	0.4103	0.3843	0.5444	0.6129	0.7008
SVM	0.4898	0.4728	0.6291	0.6890	0.7638
AAF	0.4628	0.4353	0.6095	0.6746	0.7565

It can be observed that BiLSTM outperforms Transformer. The Bidirectional Sequential Encoder achieved 68.89% and 68.15% in Macro-F1 and ACC@1 metrics, while the Transformer only got 66.86% and 66.17%. This is primarily because the average length of trajectories in the check-in dataset is very short, as indicated in Table 1, which makes Transformer unable to perform as well as it does in NLP.

5.5 Number of attention heads

To investigate the impact of the number of attention heads on the TUL solution, we compared TULSE models with different numbers of attention heads. The results are shown in Table 11. It is evident that the performance of models with 1 or 2 attention heads is significantly lower than those with 4 or more heads, indicating that multi-head attention can effectively enhance the performance of the solution. In addition, attention mechanisms with 4–10 heads exhibit similar high accuracy, and increasing the number of heads does not improve performance. This suggests that the best performance can be achieved with few attention heads, resulting in limited computation complexity.

6 Discussion

In this section, we first investigate the reason why Access Feature and Dayparting Frequency Vector can respectively express user preference and categories in subsection 6.1 and subsection 6.2. Then, subsection 6.3 shows the analysis of the computational complexity of TULSE.

6.1 Extraction of user preference

We employ the Access Feature to extract user preferences of location points. To investigate the effectiveness of this extraction, we propose an experimental approach named Average Access Feature (AAF). This approach firstly encodes every location point in trajectories to the Access Feature, which can represent user preference for location points. Then, for every trajectory, AAF calculates the average of Access Features in it to form a representation vector, which can express user preference for the trajectory. Finally, we apply Softmax on this vector to be the probability prediction vector. Notice that through the average method, AAF ignores the seriality of trajectories. Therefore, we compare AAF with SVM and LDA, which also do not take seriality into consideration, and the results are shown in Table 12. It can be observed that the AAF approach performs better than LDA but less than SVM, indicating that the Access Feature can partially extract semantic user preferences.

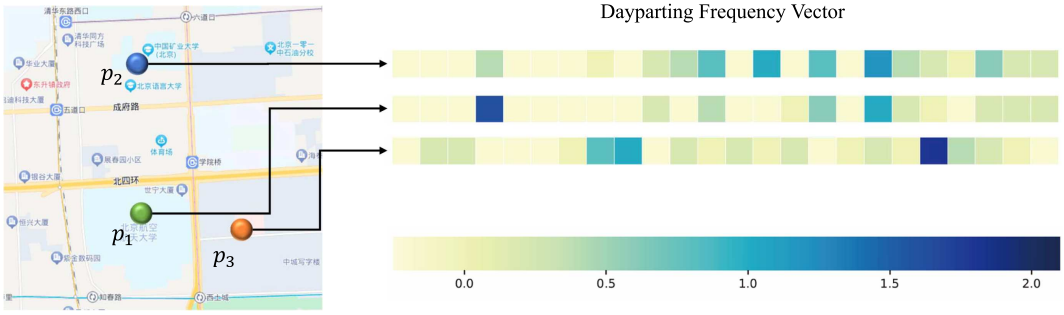

Figure 3. Heatmap comparison of three different points

Table 13. Comparison of training time on Gowalla-D

Method	TULER	TULAR	MainTUL	TULAM	TULSE
Training time(min)	1.26	1.88	82.86	3.74	2.92

6.2 Expression of category information

We employ Dayparting Frequency Vector to express the categories of location points. To assess the effectiveness of this expression, we extract the Dayparting Frequency Vectors of three real-world points on the map and visualize them in the form of a heatmap, as shown in Figure 3. Here, p_1 and p_2 represent two different colleges, while p_3 represents a hospital. It is clear that the vectors for p_1 and p_2 are relatively similar, whereas the heatmap for p_3 exhibits significantly different high-intensity areas compared to the other two points, especially in the dimensions outlined. This indicates that the Dayparting Frequency Vector can, to a certain extent, reflect the proximity of different location points in terms of their categories.

6.3 Analysis of model complexity

Suppose that the dimension of the point representation vector and output of BiLSTM are d_p and d_h, respectively, and the length of the trajectory is m. Then, the computational complexity of LSTM is $O(md_h(d_p + d_h))$, and that of a single attention head is $O(m^2 d_h)$. Therefore, the total computational complexity of the bidirectional sequential encoder with k attention heads is $O(md_h(km + d_p + d_h))$.

Furthermore, we compare the training time of different methods for 100 epochs on the Gowalla-D dataset. The results are shown in Table 13, which are obtained on one RTX 4090 GPU with i7-13700K CPU and 32G memory. It can be observed that our method is faster than the state-of-the-art methods MainTUL and TULAM. This shows that our method has acceptable time consumption.

7 Conclusion and future work

In this study, we introduce a supervised encoding-based TUL approach called TULSE. This model utilizes Supervised Spatiotemporal Encoding to generate representation vectors enriched with spatiotemporal information. It employs an attention-based BiLSTM model to more adequately extract sequence information from trajectories. We have demonstrated the effectiveness and generality of our approach through experiments on both the check-in datasets and the GPS dataset. Additionally, we explain the effectiveness of various components in TULSE through further experiments and show that our approach can achieve usability with reasonable computation complexity. Furthermore, we propose a novel metric called Hierarchical Privacy Loss (HPL), which provides a more detailed analysis of the effectiveness of the solution. It can reveal high-risk users in the dataset and assess the effectiveness of different solutions for low-risk users, making it a valuable metric for evaluating TUL solution effectiveness.

In the future, we will explore multimodal techniques to extract complex semantics from trajectories in various models. Additionally, in reality, datasets are not always static, *i.e.*, users and trajectories are

increasing, which will cause retraining for most current TUL solutions. Therefore, we will investigate the TUL task in an open environment.

Acknowledgments

Thanks Shuyu Cao for the technical support.

Funding

This work was supported by the National Key R&D Program of China (2022YFB4501500, 2022YFB4501503).

Conflicts of interest

The authors certify that they have no financial conflict of interest (*e.g.*, consultancies, stock ownership, equity interest, patent/licensing arrangements, *etc.*) in connection with this article.

Data availability statement

Our code is available at https://github.com/AIPAG/TULSE.

Author contribution statement

Chengrui Hu: Conceptualization, Writing original manuscript draft. Zheng Li: Provided suggestions on the TUL scheme and evaluation metrics, and contributed to the manuscript writing. Siyuan Wu: Offered suggestions on the TUL scheme, and assisted with manuscript revisions. Bowen Shu: Provided code and experimental support, and helped with manuscript revisions. Min Zhang: Project management and advancement, provided suggestions for manuscript revisions. Hao Li: Contribute to the research plan and manuscript writing, Corresponding author.

References

[1] Gao Q, Zhou F and Zhang K et al. Identifying human mobility via trajectory embeddings. IJCAI 2017; **17**: 1689–1695.
[2] Das G, Gunopulos D and Mannila H. Finding similar time series. In: European Symposium on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, 88–100.
[3] Yi BK, Jagadish HV and Faloutsos C. Efficient retrieval of similar time sequences under time warping. In: Proceedings 14th International Conference on Data Engineering. IEEE, 1998, 201–208.
[4] Mikolov T, Chen K and Corrado G et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.
[5] Sun T, Xu Y and Wang F et al. Trajectory-user link with attention recurrent networks. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, 4589–4596.
[6] Li H, Cao S and Chen Y et al. TULAM: trajectory-user linking via attention mechanism. Sci China Inf Sci 2024; **67**: 112103.
[7] Wang G, Liao D and Li J. Complete user mobility via user and trajectory embeddings. IEEE Access 2018; **6**: 72125–72136.
[8] Yu Y, Tang H and Wang F et al. Tulsn: siamese network for trajectory-user linking. In: 2020 International Joint Conference on Neural Networks (IJCNN). IEEE 2020; 1–8.
[9] Miao C, Wang J and Yu H et al. Trajectory-user linking with attentive recurrent network. In: Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, 2020, 878–886.
[10] Chen W, Li S and Huang C et al. Mutual distillation learning network for trajectory-user linking. arXiv preprint arXiv:2205.03773, 2022.
[11] Freudiger J, Shokri R and Hubaux JP. Evaluating the privacy risk of location-based services. In: Financial Cryptography and Data Security: 15th International Conference, FC 2011, Gros Islet, St. Lucia, February 28-March 4, 2011, Revised Selected Papers 15. Springer Berlin Heidelberg, 2012, 31–46.
[12] Zang H and Bolot J. Anonymization of location data does not work: A large-scale measurement study. In: Proceedings of the 17th Annual International Conference on Mobile Computing and Networking, 2011, 145–156.
[13] De Montjoye YA, Hidalgo CA and Verleysen M et al. Unique in the crowd: The privacy bounds of human mobility. Sci Rep 2013; **3**: 1–5.
[14] Chen L, Özsu MT and Oria V. Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 2005, 491–502.
[15] Xiao X, Zheng Y and Luo Q et al. Finding similar users using category-based location history. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2010, 442–445.

[16] Ashbrook D and Starner T. Learning significant locations and predicting user movement with GPS. In: Proceedings. Sixth International Symposium on Wearable Computers. IEEE, 2002, 101–108.
[17] Chen Z, Fu Y and Zhang M et al. The de-anonymization method based on user spatio-temporal mobility trace. In: Information and Communications Security: 19th International Conference, ICICS 2017, Beijing, China, December 6-8, 2017, Proceedings 19. Springer International Publishing, 2018, 459–471.
[18] Huo Z, Meng X and Zhang R. Feel free to check-in: Privacy alert against hidden location inference attacks in GeoSNs. In: Database Systems for Advanced Applications: 18th International Conference, DASFAA 2013, Wuhan, China, April 22-25, 2013. Proceedings, Part I 18. Springer Berlin Heidelberg, 2013, 377–391.
[19] Sadilek A, Kautz H and Bigham JP. Finding your friends and following them to where you are. In: Proceedings of the fifth ACM International Conference on Web Search and Data Mining, 2012, 723–732.
[20] Yang D, Zhang D and Zheng VW et al. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. IEEE Trans Syst Man Cybern Syst 2014; **45**: 129–142.
[21] Zhou F, Gao Q and Trajcevski G et al. Trajectory-User Linking via Variational AutoEncoder. IJCAI, 2018, 3212–3218.
[22] Cover T and Hart P. Nearest neighbor pattern classification. IEEE Trans Inf Theory 1967; **13**: 21–27.
[23] Hu X, Han Y and Geng Z. Novel trajectory representation learning method and its application to trajectory-user linking. IEEE Trans Instrum Measur 2021; **70**: 1–9.
[24] Cho E, Myers SA and Leskovec J. Friendship and mobility: User movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, 1082–1090.
[25] Zheng Y, Zhang L and Xie X et al. Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, 2009, 791–800.
[26] Zheng Y, Li Q and Chen Y et al. Understanding mobility based on GPS data. In: Proceedings of the 10th International Conference on Ubiquitous Computing, 2008, 312–321.
[27] Zheng Y, Xie X and Ma WY. GeoLife: A collaborative social networking service among user, location and trajectory. IEEE Data Eng Bull 2010; **33**: 32–39.
[28] Kingma DP and Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[29] Srivastava N, Hinton G and Krizhevsky A et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014; **15**: 1929–1958.

Chengrui Hu received a B.Eng. degree in computer science and technology from University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently pursuing a Ph.D. degree at the Institute of Software, Chinese Academy of Sciences. His research interests mainly include privacy issues of AI models, such as trajectory-user linking.

Zheng Li is a postdoctoral researcher at the CISPA Helmholtz Center for Information Security. He earned his Ph.D. in 2023 from CISPA, under the supervision of Dr. Yang Zhang, and received his Bachelor's (2017) and Master's (2020) degrees from Shandong University, supervised by Prof. Shanqing Guo. His research focuses on trustworthy machine learning, with over ten publications in top-tier security and machine learning venues.

Siyuan Wu received a B.Eng degree from the Harbin Institute of Technology, China, in 2022. He is currently pursuing a Ph.D. degree at the Institute of Software, Chinese Academy of Sciences, China. His research interests mainly include the security issues of AI models, such as membership inference attacks.

Bowen Shu received a B.Eng. degree in computer science and technology from the University of Chinese Academy of Sciences, Beijing, China, in 2021. He is currently pursuing a Ph.D. degree in cyberspace security with the University of Chinese Academy of Sciences, Beijing, China. His research interests include differential privacy and obliviousness.

Min Zhang received her Ph.D. degree from the University of Chinese Academy of Sciences, China, in 2007. She is currently an professor of the TCA Lab., Institute of Software, Chinese Academy of Sciences. Her current research interests include data security and privacy protection in cloud computing, differential privacy, and confidential computing.

Hao Li received a B.Eng degree from Xidian University, Xi'an, China, in 2005, and a Ph.D. degree in Information Security from the Institute of Software, Chinese Academy of Sciences, Beijing, China in 2011. He is currently an Assistant Professor at the Institute of Software, Chinese Academy of Sciences. His research interests focus on access control, security, and privacy in machine learning.