fuzzyCom: Privacy-Aware Trajectory Data Compression Using Fuzzy Sets in Edge Vehicular Networks

Yinglong Li, Jinyuan Shi, Dan Meng, Tieming Chen, Xinchen Xu and Fan Liu
School of Computer Science & Technology Zhejiang University of Technology, Hangzhou, China
Email:{liyinglong, 2112012219, 2112012030, tmchen}@zjut.edu.cn

2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 978-1-6654-7180-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MASS56207.2022.00091

***Abstract*—In traffic application scenarios such as real-time trajectory mining and prediction, the transmission and storage of large-scale original trajectory data engulf vehicular networks in terms of wireless transmission and time overhead, as well as easy leakage of users' sensitive location information. A novel trajectory data compression framework using fuzzy sets is proposed, which makes use of the computing and storage capabilities of vehicles and edge gateways. In our methods, raw trajectory data are compressed into fuzzy strings and then further compressed using coding techniques. Our scheme not only reduces the amount of data transmission but also protects users' location/trajectory privacy. Extensive evaluations based on real-world trajectory data sets show that our framework outperforms other baselines in terms of compression ratio, delay and information loss.**

***Index Terms*—Trajectory compression, Vehicular Network, Fuzzy sets, Privacy protection**

I. Introduction

In recent years, traffic services based on vehicle trajectory data have attracted increasing attention, such as real-time trajectory prediction and traffic flow prediction [1]- [2] in intelligent transport. Due to the rapid development of wireless communication and vehicular technologies, vehicle trajectory information can be collected and transmitted to data centers (such as Cloud Server) through GPS and OBU (On-Board Unit) [3]. However, both vehicles and edge gateways have constrained capabilities of storage, computation and wireless communication, which can not afford the communication and time cost of large-scale transmission of original trajectory data from vehicles to data centers. In addition, uncompressed data transmission also brings risks of privacy leaks.

The existing trajectory compression methods [4]- [5] strive to improve compression quality of services (QoS), such as high compression ratio, low information loss, fast compression process, etc. A naive strategy is to reduce the sampling rate of trajectory data [6]. However, the collected data obtained by this method is too sparse, and it is challenging to accurately express the positional changes between sampling points. Another method is to remove some location information that has little effect on trajectory restoration [7], which is often difficult to operate in practice. Moreover, the existing methods lack to consider the uncertainty of vehicular trajectory data, caused by OBU equipment and dynamic road conditions. Their compression calculations based on raw trajectory data are often computation-intensive, time-consuming and information-losing, which is difficult to meet the QoS in vehicular networks.

Raw vehicular trajectory data comprise massive users' sensitive location information. If the private data is directly transmitted in vehicular networks, it is easy for eavesdroppers to sniff the users' location information thus threatening their personal safety [8]. In order to protect user privacy, authentication and encryption technologies are commonly adopted to meet privacy requirements, but these methods always dramatically increase the communication burden as well as compromising the usability of trajectory data [9]- [10]. Fuzzy sets provide a good idea for analyzing and processing imprecise and uncertain data in a robust and understandable way [11]- [12]. motivated by these works, fuzzy sets are used for compressing the raw trajectory data to deal with the uncertainty of trajectory data and protect users' privacy in an energy-efficient way in edge vehicular networks.

Different from the existing work, this paper proposes a trajectory compression scheme to process the raw trajectory data at edge sides (both vehicles and gateways) using fuzzy sets. Firstly, raw trajectory data are transformed into fuzzy information, and then fuzzy information instead of raw locations are sent to edge gateways. Secondly, fuzzy information is further compressed through coding techniques. Extensive evaluations based on real-world data set validated our QoS ambitions. The main contributions can be summarized as follows:

1) A new fuzzy compression method of vehicular trajectory data is proposed. Raw trajectory data is transformed into fuzzy strings on vehicle sides, which is not only beneficial for dealing with the uncertainty of vehicular trajectory data, but also for protecting users' location privacy.

2) Improved coding compression methods combining the

The corresponding author is Tieming Chen. This research is supported in part by the Key R&D Project of Zhejiang Province (2021C01117), in part by the 2020 Industrial Internet Innovation and Development Project (TC200H01V), in part by the Major Program of Natural Science Foundation of Zhejiang Province (LD22F020002) and in part by the "Ten Thousand People Program" Technology Innovation Leading Talent Project in Zhejiang Province (2020R52011).

2155-6814/22/$31.00 ©2022 IEEE
DOI 10.1109/MASS56207.2022.00091

longest common string technique with both Huffman coding (fuzzyCom) and RLE coding (fast-fuzzyCom) are devised for further compression of fuzzy trajectory strings into binary data, which increases the compression ratio without information loss.

3) Extensive evaluations based on real-world data sets were carried out to verify the effectiveness of our framework. The experimental results show that both fuzzyCom and fast-fuzzyCom achieve higher compression ratios in desired real-time and privacy-preserving manners compared to the state-of-the-art compression methods.

II. RELATE WORK

There are several compression strategies, including original trajectory data is substituted with polylines based on trajectory geometry [6]. The principle of the first compression strategy is to retain those trajectory points that contribute significantly to the main trajectory shape. A representative work is the Douglas-Peucker algorithm [13], which recursively segments the original trajectory line by offsetting the distance between the middle point and the current baseline (connecting the head and tail points). However, the algorithm is complex and the temporal data are not considered. Top-down time ratio (TD-TR) [4] uses the time-synchronized Euclidean distance as the evaluation basis for the simplified spatial error. The BQS algorithm [14] reduces the compression time by introducing a convex hull that limits a certain number of trajectory points.

The second type of trajectory compression relies on map matching [15]- [16], which maps trajectories onto road networks. PRESS [7] uses the idea of shortest path compression to map GPS trajectory points of spatial data to road networks, and then decomposes the trajectory into multiple sub-trajectories. Finally Huffman coding is used to convert the original trajectory into binary strings. DAVT [17] compresses the distance sequence (from trajectory points to road networks), and then discretizes the distance value into several specific values, and finally clusters the distance sequence according to the road maps.

Another category is semantic trajectory compression [18] which uses semantic information such as POI (Point Of Interest), landmarks, intersections and road segments to represent the moving process of the target. Gao et al. [19] introduces a new semantic trajectory compression method by introducing a multi-resolution synchronization based clustering model to generate semantic regions of interest in a hierarchical manner. However, due to the loss of latitude and longitude information, specific point queries cannot be supported.

III. PRELIMINARIES

This section briefly introduces some basic knowledge such as trajectory data in vehicular networks, fuzzy sets and concerned privacy threats.

A. Trajectory data in Edge Vehicular Networks

Raw vehicular trajectory data collection. Vehicles periodically collect their locations through on-board GPS devices and then raw trajectory data are collected on vehicle sides. For instance, from time t_0 to t_n, the raw trajectory data of a vehicle can be described as $P = \{P_0, P_1, ..., P_n\}$. Where t_0 is the original time when the vehicle enters a new edge zone (each edge zone governed by a edge gateway), and $P_i = (x_i, y_i, t_i)$ indicates the location of a vehicle in longitude direction (x) and latitude direction (y) at time t_i. These trajectory data can be pre-processed or directly sent to the nearby edge gateways or data centres (cloud servers).

B. Fuzzy Sets

Definition 1. (Fuzzy sets) A fuzzy set F is a pair (X,$\tilde{\mu}_F$), where X is a universe of discourse and $\tilde{\mu}_F(x) : X \rightarrow [0, 1]$ ($x \in X$) is a membership function. For each $x \in X$, the value $\tilde{\mu}_F(x)$ is called membership degree of x in (X,$\tilde{\mu}$).

The membership degree in Definition 1 is between the extremes 0 and 1 of the domination of the real numbers. For instance, we define a fuzzy set "distance is long" over location variation (universe X). When the location change of a vehicle is 200 metres (longitude direction) and 230 metres (latitude direction) , and the $\tilde{\mu}_F(200, 230) = 0.8$, which means the membership degree of the location change is really a long distance to some extent. While $\tilde{\mu}_F(15, 20) = 0.1$ means the location change (15, 20) is not a long distance at all.

C. Privacy Threats

There are two main privacy risks concerned in edge vehicular network. To begin with, *Compromise Attack* including vehicle-compromise and gateway-compromise threatens the privacy of users' sensitive trajectory information. If edge nodes (both vehicles and gateways) are compromised, their crisp trajectory data face a great leakage risk without compression or encryption. Besides, *Eavesdropping Attack*: Adversaries can obtain users' private location / trajectory through sniffing or eavesdropping the data transmission among vehicles, gateways and clouds.

IV. EDGE TRAJECTORY DATA COMPRESSION

A. Fuzzy Compression of Raw Trajectory Data in Vehicles

Trajectory data prepossessing. For a privacy-preserving purpose, the raw trajectory data mentioned in Subsection III.A are not sent to local edge gateway directly. Instead, each vehicle only sends its original location (x_0,y_0,t_0) at time t_0 to its nearby edge gateway when it enters a new edge zone. Later on its location changes instead of raw location data as well as the sampling time interval are calculated and stored at the next sampling points. The location changes (also called variation) are calculated in longitude direction (x) and latitude direction (y) respectively. The location changes at time $t_i (i = 1...n)$ are calculated as: $d_i^x = x_i - x_{i-1}$ and $d_i^y = y_i - y_{i-1}$ with time interval $d_i^t = t_i - t_{i-1}$. For instance, t_0 is the first time when a vehicle enters an edge zone, and its location changes are $x_1 - x_0$ and $y_1 - y_0$ when it moves to another location at time t_1. Then the location changes (d_1^x, d_1^y) instead of raw locations are stored at time t_1.

Running examples. From time t_0 to time t_5, the trajectory of a vehicle is P=$\{P_0, P_1, P_2, P_3, P_4\}$= $\{(0.0, 0.0, 0.0), (1.02, 0.75, 0.5), (2.16, 0.25, 1.0), (3.18, 0.25, 1.5), (6.24, 0.25, 2.0)\}$, and the vehicle sends its initial location (x_0, y_0, t_0)= $(0.0, 0.0, 0.0)$ to its edge gateway at time t_0. The location variation in x direction is calculated as $d_1^x = x_1 - x_0 = 1.02 - 0.0 = 1.02$ at t_1, and the variation in y direction is $d_1^y = y_1 - y_0$= $0.75 - 0.0 = 0.75$ with time interval $d_1^t = t_1 - t_0$= $0.5 - 0 = 0.5$. Its variation sequence is obtained as (d_1^x, d_1^y, d_1^t)= $(1.02, 0.75, 0.5)$ at time t_1. Similarly, from t_2 to t_5, the variation sequences are (1.14, -0.5, 0.5), (1.02, 0, 0.5) and (3.06, 0, 0.5) respectively.

Variation sign extraction. u_i^x and u_i^y are used to represent the $i^{th}(i = 1, 2, ., n)$ variation sign in x and y direction respectively. Variation sign is either 1 when it is positive sign, or is 0 when it is negative sign. A variation after extracting its sign is called distance $|d_i^x|$ and $|d_i^y|$ in two directions. For example, when the variation in x direction is $d_1^x = 1.02$, and the variation in y direction is $d_1^y = -0.75$. Then the distance sequence after extracting variation sign can be expressed as $|d_i^x| = 1.02$, $|d_i^y| = 0.75$, $u_1^x = 1$, $u_1^y = 0$. Then the variation sequence is splitted into the distance sequence in x direction D^x=$\{|d_i^x|, ..., |d_n^x|\}$ with sign sequence U^x=$\{u_1^x, ..., u_n^x\}$ and in y direction D^y=$\{|d_i^y|, ..., |d_n^y|\}$ with sign sequence U^y=$\{u_1^y, ..., u_n^y\}$.

Base on Definition 1, we define r fuzzy sets to describe the fuzzy location variation in both x and y directions.

Definition 2. (Fuzzy sets) Let the universe X be the distance (D^x or D^y) between two adjacent positions, and define r fuzzy sets to describe the fuzzy location variation. For example, when r is 5, define fuzzy set FD^1 to describe location variation to be"very close", FD^2 "relatively close", FD^3 "medium distance", FD^4 "relatively far", FD^5 "very far", and their trapezoidal membership functions $\tilde{\mu}_{FD}^i(x)(i = 1, 2...5)$ are shown in formulas (1)-(3).

$$\tilde{\mu}_F^1(x) = \begin{cases} 1 & x < v_{FD}^1 \\ (v_{FD}^2 - x)/(v_{FD}^2 - v_{FD}^1) & v_{FD}^1 \le x < v_{FD}^2 \\ 0 & x \ge v_{FD}^2 \end{cases} \quad (1)$$

$$\tilde{\mu}_F^i(x) = \begin{cases} 0 & x < v_{FD}^1 \\ (x - v_{FD}^{i-1})/(v_{FD}^i - v_{FD}^{i-1}) & v_{FD}^{i-1} \le x < v_{FD}^i \\ (v_{FD}^{i+1} - x)/(v_{FD}^{i+1} - v_{FD}^i) & v_{FD}^i \le x < v_{FD}^{i+1} \end{cases} \quad (2)$$

$$\tilde{\mu}_F^5(x) = \begin{cases} 0 & x < v_{FD}^4 \\ (x - v_{FD}^4)/(v_{FD}^5 - v_{FD}^4) & v_{FD}^4 \le x < v_{FD}^5 \\ 1 & x > v_{FD}^5 \end{cases} \quad (3)$$

Where $\underline{x}$ and $\overline{x}$ are the lower and upper bounds of universe X. v_{FD}^i is the parameter of the above five trapezoidal membership functions. Linear trapezoidal membership functions contribute fast calculation and thus improving the compression speed.

Parameters v_{FD}^i setting. for variation sequences in both x direction and y direction, r fuzzy sets $FD^1...FD^r$ are created like formulas (1) -(3). Where the parameter v_{FD}^i of the ith fuzzy set is obtained through formula (4).

$$v_{FD}^i = \begin{cases} d_{min}^{x(y)} + \frac{(2i-1)(d_{avg}^{x(y)} - d_{min}^{x(y)})}{r} & i \le \lfloor (\frac{r}{2}) \rfloor \\ d_{avg}^{x(y)} + \frac{2(i-r)(d_{max}^{x(y)} - d_{avg}^{x(y)})}{r} & i > \lfloor (\frac{r}{2}) \rfloor \end{cases} \quad (4)$$

Where $\lfloor x \rfloor$ gets the greatest integer that is less than or equal to x. $d_{min}^{x(y)}$, $d_{max}^{x(y)}$ and $d_{avg}^{x(y)}$ is the minimum, maximum and mean variation in x direction or y direction, which periodically calculated from historical trajectory data collected by the gateway in each edge zone. When a vehicle enters an edge zone, it gets the recent parameters for its fuzzy sets through sending a request to the edge gateway.

Fuzzy partitions over location variation. The membership functions of the five fuzzy sets in Definition 2 intersect at four points: $\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$, $\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$, $\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$ and $\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$, and the universe X (variation) was divided into five non-uniform partitions: [$\underline{x}$, $\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$), [$\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$, $\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$), [$\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$, $\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$), [$\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$, $\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$) and [$\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$,$\overline{x}$] as shown in Fig.1.

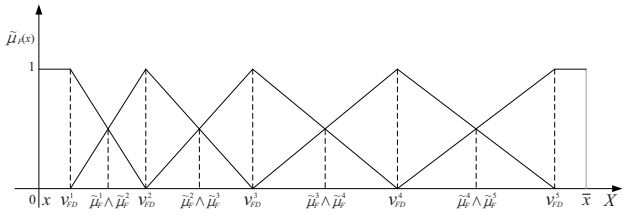


Fig. 1. Fuzzy partitions of location variation

Definition 3. (Fuzzy linguistic variable) We define a Linguistic Variable (LV) for each partition shown in Fig.1 as follows. Define linguistic variable 'N' to describe all the variation values x belonging to partition [$\underline{x}, \tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$), where $\tilde{\mu}_{FD}^i(x)$ corresponding to fuzzy set FD^1 is larger than other fuzzy sets $FD^i(i = 2, 3, 4, 5)$. Similarly, define language variables 'C', 'M', 'F' and 'G' to describe partitions [$\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2, \tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$), [$\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3, \tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$), [$\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4, \tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$) and [$\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5, \overline{x}$] respectively, as shown in TableI.

TABLE I
FUZZY PARTITIONS AND THEIR LINGUISTIC VARIABLES

LV	Fuzzy Partitions	Fuzzy sets
N	[$\underline{x}$, $\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$)	FD^1 ("very close")
C	[$\tilde{\mu}_{FD}^1 \wedge \tilde{\mu}_{FD}^2$, $\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$)	FD^2 ("relatively close")
M	[$\tilde{\mu}_{FD}^2 \wedge \tilde{\mu}_{FD}^3$, $\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$)	FD^3 ("medium distance")
F	[$\tilde{\mu}_{FD}^3 \wedge \tilde{\mu}_{FD}^4$, $\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$)	FD^4 ("relatively far")
G	[$\tilde{\mu}_{FD}^4 \wedge \tilde{\mu}_{FD}^5$, $\overline{x}$]	FD^5 ("very far")

Granularity of fuzzification (α). Fuzzy string $\{LV_1, ..., LV_r\}$ (r=$2\alpha+1$) can be obtained through Definition 2 and Table I. α is the granularity of fuzzification which is used to determine the numbers of fuzzy sets, partitions and Lvs. Generally speaking, the larger the α, the more accurate the fuzzy information is described, which is helpful

for reducing the information loss. However bigger α causes more involved LVs and more complicated compression and decompression. Besides, the larger α has a negative impact on both compression ratio and real-time performance.

B. Further Compression in Edge Gateways

After obtaining the fuzzy strings in both x and y directions, the fuzzy data is further compressed by replacing common sub-strings and encoding techniques. Firstly, there are a large number of repeated sub-strings in the obtained fuzzy strings in both x and y directions, so a specific linguistic variable can be used to replace the common sub-strings to obtain new fuzzy strings. Secondly, fuzzy strings can be further compressed either using Huffman coding or RLE (Run-Length Encoding) [20] techniques.

Common substring-based compression. Edge gateways collects the fuzzy strings S^x=$\{S_1^x, ..., S_m^x\}$ and S^y=$\{S_1^y, ..., S_m^y\}$ sent by each vehicle. In order to maximally substitute the repeated sub-strings in fuzzy strings, the longest common substrings in both S^x and S^y are calculated according to the definition of k-common substring problem in WikiMili [21].

Each the longest common sub-strings is replaced with a specific LV in fuzzy strings on edge gateway side to achieve maximum compression. For instance, when there are four fuzzy strings: S_1=MFCCMNFGC, S_2=GCMCMNFMC, S_3=FCFCMNFCN and S_4=FCCMFNMNF. The longest common substring is CMNF and then CMNF is represented by a linguistic variable L, and the four new fuzzy strings are $S_1^{'}$=MFCLGC, $S_2^{'}$=GCMLMC, $S_3^{'}$=FCFLCN and $S_4^{'}$=FCLNMNF.

We further compress the fuzzy trajectory data after common substring-based compression using Huffman coding (fuzzyCom) and RLE coding (fast-fuzzyCom) respectively. fuzzyCom has better compression ratio. While the time complexity of fast-fuzzyCom is O(n), which is much lower than that of fuzzyCom (O(n^2)), which makes fast-fuzzyCom more suitable for real-time scenarios.

fuzzyCom. Huffman coding is a binary code with variable coding length. The main idea is that the greater the frequency of a LV, the shorter the corresponding code. When the frequency of each LV appearing in a fuzzy string is calculated, and then a Huffman tree can be established, which makes Huffman binary coding accordingly. For instance, when there are four LVs (CMFL) in a fuzzy string and their frequency are 0.1, 0.4, 0.3 and 0.2 respectively. Then the corresponding Huffman tree can be obtained as M (0), F(10), C (110) and L (111). The Huffman binary coding sequence of a fuzzy string (MFCML) is 0101100111.

fast-fuzzyCom. RLE (Run Length Encoding) is a efficient and lossless statistical coding technique for compression. In RLE, raw data is regarded as a linear sequence, and it contains two categories: one is continuous repeated data block, and the other is non-repetitive data block. The compression strategy of RLE for continuous repeated data block is to use a byte (we call it the data multiplicity) to represent the number of repetitions of the data block followed by the repeated data block. For instance, a data sequence AAAAA, which occupies 5 bytes before uncompressed, and becomes 5A after RLE compression, occupying only 2 bytes. The compression method for non-repetitive data sequences is similar, except that the number is often 1 before the non-repeated data block. Fuzzy strings can be further encoded using RLE coding method and its time complexity is O(n), which is much lower than O(n^2) of Huffman coding. Therefore, RLE based fast-fuzzyCom is more suitable for senarios with high real-time QoS requirement.

A running example. Fig.2shows an example of both fuzzyCom and fast-fuzzyCom. To begin with, the raw trajectory data of a vehicle is transformed into distance sequences, sign sequences both in x and y directions, as well as time sequence. Then, the distance sequences are transformed into a fuzzy string through fuzzification. In addition, the fuzzy string is further compressed via LCS compression and coding techniques.

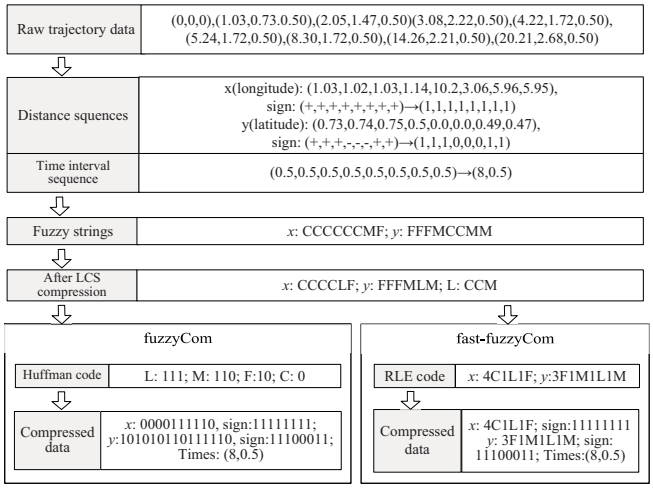


Fig. 2. Running examples of fuzzyCom and fast-fuzzyCom

Compression of time sequences. The sampling period of trajectory data is usually fixed, so RLE coding is a preferable technique to compress time sequences $D_1^t, .., D_m^t$. For example, when sampling period t is 0.5, the time sequence after six periods is (0.5, 0.5, 0.5, 0.5, 0.5, 0.5), then it can be compressed to be (0.5, 6) via RLE technique.

Privacy protection analysis. In both fuzzyCom and fast-fuzzyCom, raw trajectory data is transformed into fuzzy information before being sent to gateway, which can resist eavesdropping attacks from malicious vehicles during data transmission between vehicles and gateways. Besides, fuzzy information futher compressed into binary data either using Huffman coding technique in fuzzyCom or using RLE coding technique in fast-fuzzyCom, which has excellent immunity against trajectory privacy leakage via compromising attacks both on gateway and data centre sides.

C. Overall framework

The overall framework of the proposed trajectory data compression in edge vehicular networks is shown in Fig.3. On the vehicle side, vehicles collect their location periodically time through on-board GPS devices, and calculating their moving variations in both x-direction (longitude) and y-direction (latitude) respectively. Then their moving variations are divided into distance sequences D_i^x in x-direction and D_i^y in y-direction and their sign sequences U_i^x and U_i^y. Then D_i^x and D_i^y are transformed into fuzzy strings according to Definition 2 and Table I. During the fuzzification, parameters for fuzzy membership functions are updated periodically through statistical calculations in edge gateways. D_i^t is compressed through RLE coding technique on vehicle side as well. The fuzzy strings, encoded time sequences, U_i^x and U_i^y are sent to V_i's nearby gateway along with each vehicle's initial location. On the edge gateway side, the received fuzzy strings sent by each vehicle are further compressed into binary-coded information using both longest common substring (LCS) (SA_IS algorithm [22] is recommended for calculating the suffix array due to its lower time complexity) and coding techniques (Huffman coding for fuzzyCom and RLE coding for fast-fuzzyCom). More details are shown in Algorithm 1.

Algorithm 1 fuzzyCom & fast-fuzzyCom

Input: Original trajectory data of vehicle $P = \{P^1, P^2, ..., P^m\}$, $P^i = \{(x_0, y_0, t_0), ...(x_n, y_n, t_n)\}, i = 1, 2..m$.
Output: Compressed trajectory data (binary data).
1: **for** each vehicle V_i $(i = 1, ..., m)$ **do**
2: V_i calculates the variation sequences d_i^x, d_i^y and d_i^t of P_i.
3: The variation sequences are divided into distance sequences (D_i^x and D_i^y), sign sequences (U_i^x and U_i^y) and time sequences (D_i^t) respectively;
4: D_i^x and D_i^y are transformed into fuzzy strings according to Definition 2 and Table I, and these fuzzy strings as well as U_i^x and U_i^y are sent to V_i's nearby gateway.
5: D_i^t is compressed through RLE coding technique and sent to V_i's nearby gateway as well;
6: **end for**
7: After a gateway G_g has received all the transformed trajectory data from V_i $(i = 1, .., m)$ *// before V_i leaving G_g's edge zone*
8: G_g compresses V_i's fuzzy strings using LCS strategy in Subsection IV.B.
9: *fuzzyCom*: Huffman coding technique is used to further encode the fuzzy strings and return the binary data;
10: *fast-fuzzyCom*: RLE coding technique is used to further encode the fuzzy strings and return the binary data;

V. Experimental Evaluation

A. Experiment Setup

We conducted experimental evaluations based on real-world data set using Python platform (CPU: AMD Ryzen 7 4800H, memory: 16 GB, OS: Windows 10). The main parameters in experiments were set as below, shown in the table II.

Data set. The real-world data set used in our experimental evaluation was the NGSIM (Next Generation Simulation) [23], which consists the trajectory data of all the vehicles on US-101, peachtree and other roads. In our evaluation, the trajectory data of Highway US-101 and Peachtree were used.

TABLE II
Simulation Parameter Setting

Parameters	Values
Sampling rate	0.2 s
Fuzzy granularity α	1, 2, 3, 4, 5, 7
Length of road segment	10, 15, 20, 25, 30, 35, 40
Common substring ratio k	1, 2, 3, 4, 5, 6, 7
Load segments	Highway US-101 and City Road Peachtree [23]

The length of US-101 was approximately 640 metres including 5 main lines throughout the segment. Compared with US-101 expressway, the average speed of the vehicles on Peachtree (urban road) is slower and vehicles on it shift more frequently due to its complex road conditions.

Baselines. We compared our method with three state-of-the-art baselines as follows.

- FBQS [14]. FBQS is an online compression algorithm which reduces the compression time by introducing a convex hull that limits the number of trajectory points.
- PRESS [7]. PRESS is a map matching based compression algorithm. It maps trajectory data to a road network, and uses the idea of Shortest Path Compression to filter some unnecessary edges of the mapped network.
- DP [13]. DP (Douglas-Peucker) is an offline compression algorithm. It calculates the offset distance and recursively segment the original trajectory. When the maximum offset distance of a trajectory line segment is less than the preset geometric error η, the trajectory segment is simplified and fitted to its corresponding reference line.

Evaluation metrics The following four metrics were used in our experimental evaluation.

- Compression ratio (cr): cr is defined as the ratio of the storage space occupied by the original data ($space(Traj)$) to the storage space occupied by the compressed data ($space(Traj)$).

$$cr = \frac{space(Traj)}{space(Traj^{'})} \tag{5}$$

- Compression time. Compression time is used to measure the efficiency of time overhead of compression.
- Maximum TSND (Time synchronized network distance): For a distance sequence, the maximum TSND is used to measure the information loss after compression [24], as shown in formula (6).

$$TSND = Max_{i=1}^{p}(|P_i^{'} - P_i|) \tag{6}$$

Where $|P_i^{'} - P_i| = \sqrt{(x_i - x_i^{'})^2 + (y_i - y_i^{'})^2}$ represents the distance between location $P_i^{'}$ (after decompression) and P_i (before compression) at time t_i.

B. Compression Effect Evaluation

We evaluated the impact of fuzzy granularity α on compression ratio and compression time on both highway US-101 and Peachtree. The length of road segment was 30 metres and the common substring ratio K was 2. The experimental results are shown in Fig.4. Fig.4(a) shows that the compression

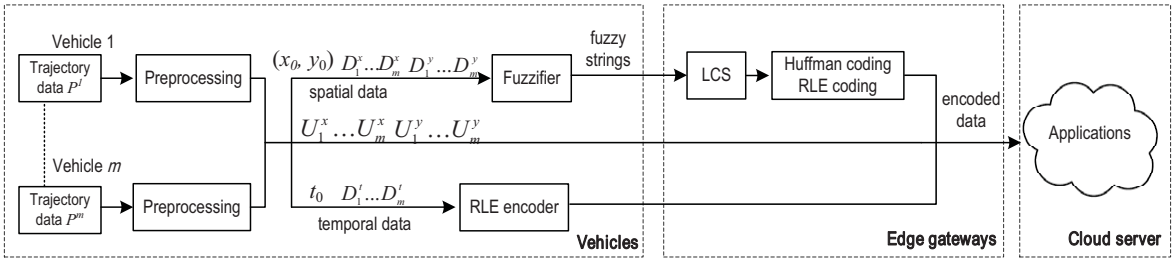
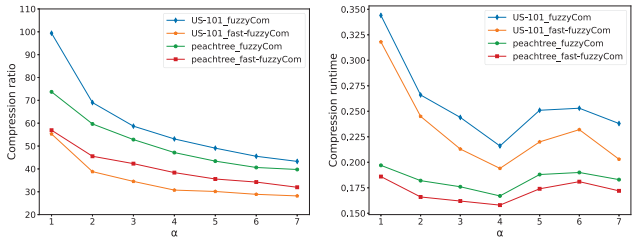


Fig. 3. Trajectory compression in the edge environment

ratio decreased with the increase of α. When α varied from 1 to 2, the compression ratio of both US-101 and peachtree roads dropped sharply. Because the larger the α, the more fuzzy linguistic variables involved, which leads to a lower compression ratio. The compression ratio of FuzzyCom of US-101 was greater than that of Peachtree with the same α, while fast-fuzzyCom was opposite, because the average vehicle speed of urban Peachtree was lower than that of US-101. There are more continuous repeated characters in fuzzy strings of Peachtree with the same α, so the compression ratio of fast-fuzzyCom on Peachtree was greater than that of US-101. The compression time of US-101 was larger than that of Peachtre with the same α, as shown in Fig.4(b). When α was 4, the number of fuzzy sets (linguistic variables) is optimal, which is beneficial to the computation of fuzzification, LCS and coding, and thus led to the lowest time overhead.



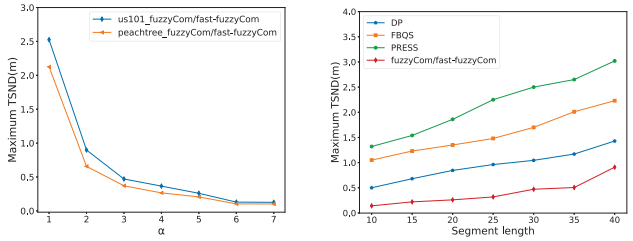
(a) (b)

Fig. 4. Compression ratio (a) and compression time (b) with different α on US-101 and Peachtree

C. Information Loss

We examined how parameters, such as α and road segment length, influence information loss with fixed k (was set to be 2) in LCS. Fig.5(a) shows how α impacted the information loss of both fuzzyCom and fast-fuzzyCom.

Because both RLE coding and huffman coding are lossless compression methods, the information losses of fuzzyCom and Fast-fuzzyCom were the same. When α was 1, the number of fuzzy languages was the smallest, and the maximum TSND was 2.5 meters (biggest). As α increased, the maximum TSND dropped sharply at the beginning, and then slowly declined, as shown in Fig.5(a). Fig.5(b) shows the impact of different segment lengths on information loss. As segment



(a) Information loss with different α (b) Information loss with different segment lengths

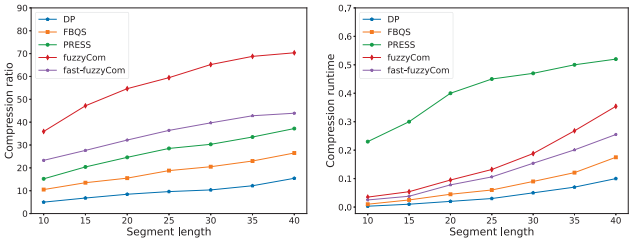
Fig. 5. Comparison of information loss

length increased, the information loss also rose gradually. The information losses of fuzzyCom and fast-fuzzyCom were smaller than other benchmark methods due to the fuzzy sets combined with lossless coding techniques used in our scheme.

D. Comparison with Baselines

By comparing the proposed method with benchmark methods (DP, FBQS and PRESS), the impact of different road segment lengths on compression ratio and compression time were assessed respectively. As can be seen from Fig.6(a), with the increase of segment length, the compression ratio also increased gradually, and the compression ratio of fuzzyCom was much larger than other baselines. With the increase of segment length, the compression time gradually grew, as shown in Fig.6(b). The compression time of fuzzyCom was slightly larger than that of DP and FBQS, but much smaller than that of PRESS. Because PRESS requires map matching, thus makes the calculation time be relatively large. Fig.6 shows that our methods achieve desirable compression ratio, compression time, as well as the balance between them.

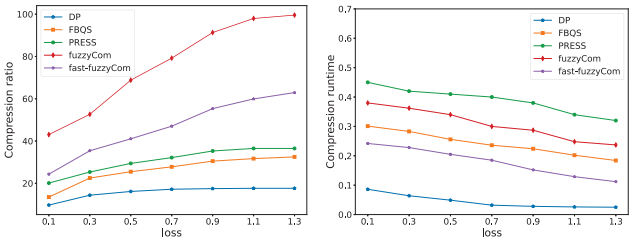
The comparison evaluation in terms of compression ratio and compression time with different information losses were assessed. Different information losses were obtained through fixing the thresholds of DP, FBQS and PRESS, and fuzzy granularity α. As can be seen from Fig.7(a), the compression ratio increased gradually when information loss grew. The compression ratios of both fuzzyCom and fast-fuzzyCom were much larger than other benchmark methods with the same information loss. With the increase of loss, the compression time of DP, FBQS, PRESS, fuzzyCom and fast-fuzzyCom



(a) (b)

Fig. 6. Compression ratio (a) and compression time (b) with different road segment lengths

gradually rose, as shown in Fig.7(b). When the thresholds of DP, FBQS and PRESS rise, the calculation time for reserving the key trajectory points is reduced. Whereas, the increase of information loss in fuzzyCom and fast-fuzzyCom as well as the drop of α both make the time overhead on LCS calculation and replacement smaller, resulting in less compression time.



(a) (b)

Fig. 7. Compression ratio (a) and compression time (b) with different segment lengths

VI. Conclusion

This paper proposes a novel trajectory data compression framework using fuzzy sets in edge vehicular networks. Raw trajectory data is transformed into fuzzy information on vehicle side before sending to local gateways, which can resist eavesdropping attacks from malicious vehicles during data transmission between vehicles and gateways. Besides, fuzzy information further compressed into binary data either using Huffman coding technique in fuzzyCom or using RLE coding technique in fast-fuzzyCom, which can resist compromising attacks on gateway sides. Besides privacy protection, our fuzzy compression framework has preferable QoS performance in terms of compression ratio, compression time and information loss based on real-world data sets when compared with some benchmark methods.

References

[1] Nitin Kamra, Hao Zhu, and et al. Multi-agent trajectory prediction with fuzzy query attention. *Advances in Neural Information Processing Systems*, 33:22530–22541, 2020.

[2] Xiaoyang Wang, Yao Ma, Yiqi Wang, and et al. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of The Web Conference 2020*, pages 1082–1092, 2020.

[3] X. Fan, W. Cai, and J. Lin. A survey of routing protocols for highly dynamic mobile ad hoc networks. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1412–1417. IEEE, 2017.

[4] Yan Ding, Chao Chen, Xuefeng Xie, Kai Liu, and Liang Feng. An online trajectory compression system applied to resource-constrained gps devices in vehicles. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–3. IEEE, 2018.

[5] Penghui Sun, Shixiong Xia, Guan Yuan, and Daxing Li. An overview of moving object trajectory compression algorithms. *Mathematical Problems in Engineering*, 2016, 2016.

[6] Sijing Liu, Gang Chen, Long Wei, and Guoqi Li. A novel compression approach for truck gps trajectory data. *IET Intelligent Transport Systems*, 15(1):74–83, 2021.

[7] Renchu Song, Weiwei Sun, Baihua Zheng, and Yu Zheng. Press: A novel framework of trajectory compression in road networks. *arXiv preprint arXiv:1402.1546*, 2014.

[8] Roger Piqueras Jover. The current state of affairs in 5g security and the main remaining security challenges. *arXiv preprint arXiv:1904.08394*, 2019.

[9] Azeem Irshad, Muhammad Usman, Shehzad Ashraf Chaudhry, Husnain Naqvi, and Muhammad Shafiq. A provably secure and efficient authenticated key agreement scheme for energy internet-based vehicle-to-grid technology framework. *IEEE Transactions on Industry Applications*, 56(4):4425–4435, 2020.

[10] Ling Xing, Xiaofan Jia, Jianping Gao, and Honghai Wu. A location privacy protection algorithm based on double k-anonymity in the social internet of vehicles. *IEEE Communications Letters*, 25(10):3199–3203, 2021.

[11] Yinglong Li, Fan Liu, Jiaye Zhang, Tieming Chen, and et al. Privacy-aware fuzzy skyline parking recommendation using edge traffic facilities. *IEEE Transactions on Vehicular Technology*, 70(10):9775–9786, 2021.

[12] Yinglong Li, Weiru Liu, Yihua Zhu, and et al. Privacy-aware fuzzy range query processing over distributed edge devices. *IEEE Transactions on Fuzzy Systems*, 30(5):1421–1435, 2022.

[13] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.

[14] Jiajun Liu, Kun Zhao, Philipp Sommer, and et al. A novel framework for online amnesic trajectory compression in resource-constrained environments. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2827–2841, 2016.

[15] Qi Cao, Gang Ren, Dawei Li, Haojie Li, and Jiangshan Ma. Map matching for sparse automatic vehicle identification data. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[16] Marko Dogramadzi and Aftab Khan. Accelerated map matching for gps trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[17] Chao Chen, Yan Ding, Suiming Guo, and Yasha Wang. Davt: an error-bounded vehicle trajectory data representation and compression framework. *IEEE Transactions on Vehicular Technology*, 69(10):10606–10618, 2020.

[18] Chongming Gao, Zhong Zhang, Chen Huang, Hongzhi Yin, Qinli Yang, and Junming Shao. Semantic trajectory representation and retrieval via hierarchical embedding. *Information Sciences*, 538:176–192, 2020.

[19] Chongming Gao, Yi Zhao, Ruizhi Wu, Qinli Yang, and Junming Shao. Semantic trajectory compression via multi-resolution synchronization-based clustering. *Knowledge-Based Systems*, 174:177–193, 2019.

[20] V Pandimurugan, J Amudhavel, M Sambath, et al. Hybrid compression technique for image hiding using huffman, rle and dwt. *Materials Today: Proceedings*, 2022.

[21] Wikipedia. Longest common substring problem. https://wikimili.com/en/Longest_common_substring_problem, 2021.

[22] Ge Nong, Sen Zhang, and Wai Hong Chan. Linear suffix array construction by almost pure induced-sorting. In *2009 data compression conference*, pages 193–202. IEEE, 2009.

[23] Ngsim: https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm. Access date: Oct. 16, 2021.

[24] Xiaochun Yang, Bin Wang, Kai Yang, Chengfei Liu, and Baihua Zheng. A novel representation and compression for queries on trajectories in road networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(4):613–629, 2017.