CWEmd: A light-weight Similarity Measurement for Resource Constraint Vehicular Networks

Cheng Qiao*†, Kenneth N. Brown†, Yong Zhang ‡ and Zhihong Tian *
* Cyberspace Institute of Advanced Technology, Guangzhou University, China
† Insight Centre for Data Analytics, Department of Computer Science, University College Cork, Ireland
‡ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
mcheng.qiao@gmail.com, k.brown@cs.ucc.ie, zhangyong@siat.ac.cn, tianzhihong@gzhu.edu.cn

***Abstract*—Generating an accurate machine learning (ML) model is of great importance for the Internet of Vehicles (IoV). However, obtaining such a model is challenging due to the fact that sub-groups of in-network vehicles receive data from different resources. A worthwhile investment then would be identifying those groups before inferring models. Similarity metrics are widely used to distinguish different groups. However, the efficiency of most existing similarity measurements is at the cost of increased computational complexity and decreased accuracy, making them unsuitable for IoV's stringent conditions. To address this issue, we propose a computationally efficient method to measure the similarity of different vehicles, where a simplified version of Earth Mover's Distance (EMD) is adopted. This distance metric is then embedded into a distributed clustering algorithm to learn the global pattern for vehicular systems. Our algorithm's overall performance is measured using an Asynchronous Message Delay Simulator. Compared to the best algorithm of the state-of-the-art, our proposed algorithm converges slightly slower (by less than 1%) but improves the clustering accuracy by as much as 20% with synthetic data. Additionally, real-world data collected from Vehicles validates the efficiency of our proposed algorithm.**

***Index Terms*—Internet of Vehicles, Earth Mover's Distance, Similarity measurement, Distributed algorithm,**

I. Introduction

INternet of Vehicles (IoV) has gained increasing attention for the benefit of improving driver comfort, traffic efficiency and road safety [1–3]. There are two main types of IoV applications: Safety-related and Non-safety-related [4]. In Safety-related applications (i.e., cooperative collision avoidance and life-threatening traffic conditions), it is crucial to provide a delay-tolerant service. Intuitively, the chances of having a traffic accident are reduced substantially if the altering of traffic avoidance service is informed in advance. On the other hand, Non-safety-related applications aim to enhance traffic efficiency (i.e., by providing weather information, traffic light updates, and gas station locations on current traffic). Nevertheless, providing such services poses a significant challenge due to the stringent conditions of IoV, including limited resources and unreliable bandwidth [1].

Manuscript received September 15, 2022; revised March 26, 2023 and May 04, 2023; accepted May 22, 2023 (*Corresponding author: Zhihong Tian*).

In many practical applications (i.e., traffic jam warnings), there are sub-groups of agents observing data from different phenomena, which are "patterns" to be identified in the network. Simply aggregating models from different patterns until approaches the approximate global model leads to a degradation in performance inevitably [5, 6]. A worthwhile investment then would be identifying those patterns before inferring similar models from agents in the same sub-group. Thus, a similarity measurement, which considers the salient characteristics of IoV, such as limited bandwidth and constrained resources, is required.

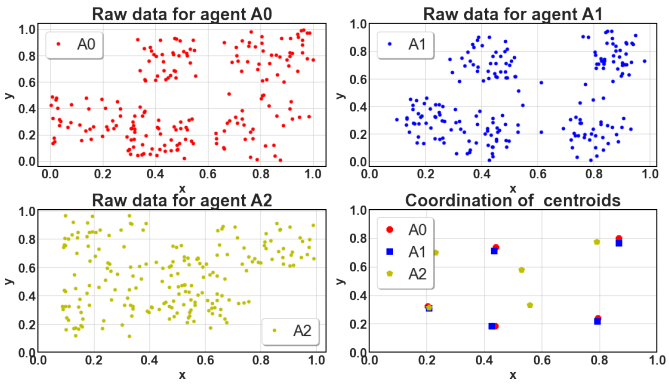

Fig. 1: The first 3 figures show the raw data for agent A_0, A_1 and A_2 (left to right). The last figure shows the cluster centroids for each agent. Obviously, it is difficult to decide which agents are similar (and so belong to the same group) and which are dissimilar (and so belong to different groups) only with the raw data points. However, we could know the groups of agents by looking at the centroids instead (see the last figure).

Earth Mover's Distance (EMD) is widely used as the similarity measurement due to its robustness to random noise, shape-awareness at a global level, and insensitivity to changes in topology [7–11]. Furthermore, it can handle vectors with different arbitrary dimensions. However, the heavy computation cost severely hinders the applicability of this distance in large-scale applications. The recent introduction of Sinkhorn Distance makes it possible [12], and the proposed approximate Sinkhorn algorithm converges in near-linear time [13]. Nevertheless, the prominent regularized version of Sinkhorn

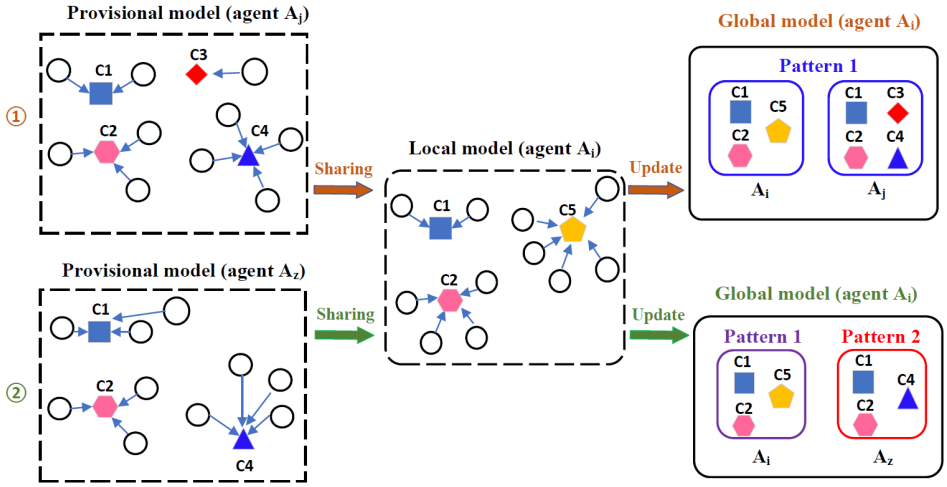

Fig. 2: An toy example for network with multiple patterns. Suppose agent A_i computes its local model and receives a provisional model from neighbours, she has to identify the patterns from these two provisional models. Agent A_i estimates that the local model from neighbour A_j is almost the same, so only one pattern is identified (the top row). However, two patterns are estimated when local model from neighbour A_z is received (the bottom row).

approximation is less accurate, making it hard to apply in most applications [14].

To tackle these problems, in this paper, we propose a lightweight EMD-based similarity measurement to distinguish different patterns, which caters to the stringent conditions of IoV. Specifically, a clustering algorithm is used to mine the underlying pattern from the raw data points of agents and an adaptive EMD metrics is applied to the cluster models. Clustering algorithms are fast, computationally efficient and highly flexible with unlabeled raw data. Moreover, it is much easier to distinguish agents that are qualitatively dissimilar from each other once an underlying pattern is established (see Figure 1 for an example). Then, agents compare the cluster models sent by neighbours with their own cluster models to identify the patterns in the network by measuring the similarity between them (see Figure 2 for an example).

Since the whole dataset is represented by the cluster model instead, it will substantially reduce the computation cost of similarity measurement. Another benefit of using EMD is that it solves the initialisation problem caused by clustering algorithms. It is possible that agents will end up with different estimations about the hyperparameter $\boldsymbol{k}$ since the raw data collected is highly varied. The globally shape-aware nature of EMD allows each agent to carefully choose its initial centroids based on its raw data (and so $\boldsymbol{k}$ across the network is not consistent), which greatly improves the accuracy of local provisional models. Therefore, it is reasonable to efficiently compute an accurate global model using these two models in the context of heterogeneous resource data.

The proposed similarity measurement is then embedded into a clustering algorithm to learn the pattern of the network. To measure the performance of the proposed methods, different distance metrics are embedded into the cluster algorithm, and the performance of the network is measured. An asynchronous message delay simulator is used to evaluate the performance, and empirical results show that our new methods can retain a high clustering accuracy (relative to centralised and ground truth baseline), and improve clustering accuracy against pattern by up to 20%. Compared to algorithms with standard EMD as a distance metric, the clustering accuracy obtained by the new approaches is similar. The proposed methods are only marginally slower (less than 1%) than the best algorithm of existing methods, but twice faster than that with standard EMD as distance metric. In addition, GPS trajectory data (WGS84 geodetic system) from the Transport Commission of Shenzhen Municipality is used to evaluate the performance of the proposed algorithm. The main contributions of this paper can be summarized as below:

- We propose a computationally efficient EMD-based method to measure the similarity of agents, which aims to identify different patterns in the resource-constraint vehicle networks. This method works well even the resource data is heterogeneous, which makes it a potential method for various applications in IoV.
- To extensively test the performance of proposed metrics, we propose an improved version of the Rand Index and F-measure to measure the performance.
- Compared to the state-of-the-art, experimental results show that the proposed algorithm mitigates the high computation cost and reduces the convergence time while retaining high clustering accuracy. In addition, the efficiency is reinforced by GPS trajectory data (WGS84 geodetic system) from the Transport Commission of Shenzhen Municipality.

The rest of this paper is organized as follows. We describe the related work in next section, then define the overall framework. After that, the experimental set-up and underlying assumption are described. In the last section, we discuss the empirical results of the experiments.

II. RELATED WORKS

Some existing works on VNs focus on pattern detection, with the aim of understanding urban phenomena, such as monitoring general traffic jams and studying the regular routines of citizens. In addition, security protections have attracted more and more attention since safety-related applications is of great importance in VNs. Another series of studies concentrate on understanding the characteristics of VNs. For instance, the communication loss of IEEE 802.11p-based Dedicated Short Range Communications is investigated in [15]. In this section, we outline the clustering algorithms used in pattern detection, describe the methods to measure the similarities, and introduce the distributed learning in wireless networks.

A. Pattern detection and security

Clustering algorithms, which group similar instances into the same cluster and separate dissimilar instances into different clusters, are widely adopted to mine traffic patterns in IoV[16–20]. Nezerenko et al. distinguished countries with similar transportation trends using Hierarchical Cluster Analysis (HCA) based methodology. A real world traffic data (from 2004 to 2011) collected from the Baltic Sea Region (BSR) was used to evaluate the performance and identify the main reasons leads to these trends [17]. Cluster analysis is shown to have the potential to improve the high quality services, which is a crucial goal for the public transport administrations [18]. Moreover, a density-based clustering methods is used to identify the complete taxi-cab trips in Beijing City [19]. Data mining methods, including agglomerative hierarchical clustering and k-nearest neighbor method, are used to model the dynamic traffic flow [20]. To summarise, clustering algorithms has the potential to identify the underlying patterns for traffic flows.

Privacy and Security are another important concerns for IoV. Since agents in ITS are connected, attackers may have physical access to a subset of system. Some works are proposed to prevent potential attackers for system access and detect malicious activities within the system [21, 22]. For instance, a mechanism, dubbed Footprint, is proposed to detect the Sybil attacks in Urban Vehicular Networks in [23]. An excellent survey of various anomaly detection technologies in IoV is presented in [24].

B. Similarity measurement

The concept of similarity is at the very heart of scientific field, such as artificial intelligent, mathematics and medicine. It is mainly used to distinguish some agents that are different from others and generally there are two main methods: Feature based [25] and Distance based [26] methods. Feature based similarity is widely used in recommendation system and e-commerce. It was used to describe the preference in recommendation systems [27]. Based on this preference, it recommends related products to a potential customer. An extensive study of various methods to measure the similarity, including Pearson correlation, Cosine similarity, Spearman rank coefficient and Jaccard similarity, are explored in this excellent survey [28].

Distance metrics (i.e., L1- and L2-norm Euclidean Distance) is one of the most influential methods to measure the similarity [29, 30]. Generally, it assumes that vectors to compare are in the same dimensional space. For more details, refer to this excellent survey on distance measure [31]. However, the vectors are not necessarily in the same dimensional space in some other applications. For instance, infer the common preference of users based on the rating of a given list of items. The rating for some items may be missing for uncertain reasons. It makes little sense to compare the similarity computed in different dimensional spaces. Sun et al. proposed a similarity measure, which normalizes the distance between two vectors in multidimensional space, to compute how similar the preference of two users are [32]. However, the final similarity is largely depending on the maximum distances in the same space.

In the context of vectors with arbitrary dimensions, Earth Mover's Distance (EMD) [33] is introduced. It is shown that this metric is insensitive to noise and globally shape-aware. However, it is only considered in the Computer Vision community for its expensive cost, with a computation cost as high as $O(n^3 logn)$ for n signatures.

There are several works aiming to reduce the high computation cost of EMD, while retaining the advantages. Generally, an approximate EMD, rather than standard EMD, is obtained instead. Hirdhonkar et al. proposed that the approximate EMD could be obtained by measuring the difference between two transformed descriptors, which transfers the data array by a wavelet function [34]. With this method, the computation cost could be decreased to $O(n)$. The authors show that the error, which is denoted as the ratio of this approximated EMD to the standard EMD, is bounded. However, this error could be as high as 7. Pele and Werman showed that the computation cost could be reduced to $O(n^2 logn)$ if intermediate destinations are allowed [35]. The number of edges could be decreased by an order of magnitude. However, a threshold distance has to be pre-defined and it is too complex for the set-up process. The recent approximate Sinkhorn distance converges in near-linear time, but it is less accurate [12, 13], making it hard to apply in most applications [14].

C. Distributed learning in wireless network

Generally, there are three main steps for the distributed learning: 1) **Local learning step**. Each agent computes a local representation which could describe the local raw data as accurate as possible; 2) **Update step**. Each agent refines its provisional model if local models from neighbours is delivered; 3) **Convergence step**. An agent stops updating their local models when the pre-defined convergence conditions are met. Based on how to compute local pattern, we can classify these algorithms into two sorts: (1) Supervised: Neural networks with several layers; (2) Unsupervised: Clustering algorithms with structure-free network.

Supervised methods. Deep neural networks (DNN) have proved to be remarkably effective for many non-convex machine learning tasks. Tens of thousands parameters are trained by fitting massive raw examples, and they are adjusted by

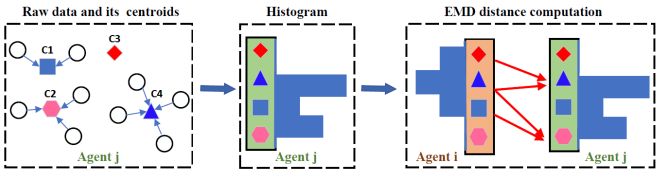

Fig. 3: Weighted EMD between agent i and j over their cluster centroids. Note that the number of centroids for these two agents is not necessarily consistent, which solves the problem of clustering algorithm in turn. Agents could

minimizing the loss function ζ with the mini-batch stochastic gradient descent (SGD) algorithm [36]. Federated Learning is a gradient-based distributed framework, and widely used in many applications for its efficiency and privacy benefit. However, it requires massive labeled raw samples to train the parameters and huge computation cost, which is obviously not cater for the stringent condition of IoV [37].

Unsupervised methods. Given the fact that the shortage of labeled data is prominent in various applications, clustering algorithms (i.e., K-means) are adopted to infer local models [38–41]. Each agent describes its local data as centroids and corresponding data points associated with them. This summary description is then shared with neighbours in a synchronised way. An agent has to wait until updates from all of its neighbours is delivered. However, this methods fails to converge if the clusters over different agents have significantly different shapes. An asynchronous distributed clustering algorithms for resource constraint network is proposed in [42]. K-means and GMM are used to compute the local model, and the shape of clusters are further explored by bounding boxes and standard deviations. Empirical result show that more informative description of clusters reduce the network overhead, improve the clustering accuracy (relative to the centralised clustering algorithm) and decrease the convergence time.

III. Overall Framework

In this section, we describe the overall framework. Whenever a cluster model is received, an agent has to compare this model with its local model and run a similarity test on these models. If multiple patterns have been detected by computing the similarity, it is necessary to compute the global view for each pattern. The basic procedure for learning the global model for multiple patterns network comprises: 1) the similarity between agents is measured, and those similarities are used to construct a similarity matrix; 2) a hierarchical clustering is then applied to detect different patterns; and 3) an asynchronous clustering algorithm is used to learn the global pattern for each pattern, as discussed below.

A. Similarity measurement

To mitigate the high computation cost of EMD, we proposed two computational efficient EMD based method to measure the similarity between different agents. *Our intuition is that, the computation cost could be significantly reduced if the massive raw data are described as some local efficient representations and the EMD metrics are applied to those representations.* As a first step, we cluster the raw data into several groups by K-means and represent the agent as centroids of those groups.

First, we propose a weighted EMD metric. The EMD metric is applied to the cluster centroids. Figure 3 shows the general idea for agent j to measure the similarity to agent i when only local representations are given. Agent i converts its local cluster model (a.k.a, centroids and corresponding size) to a histogram, then the standard EMD is applied to measure the distance between these histograms.

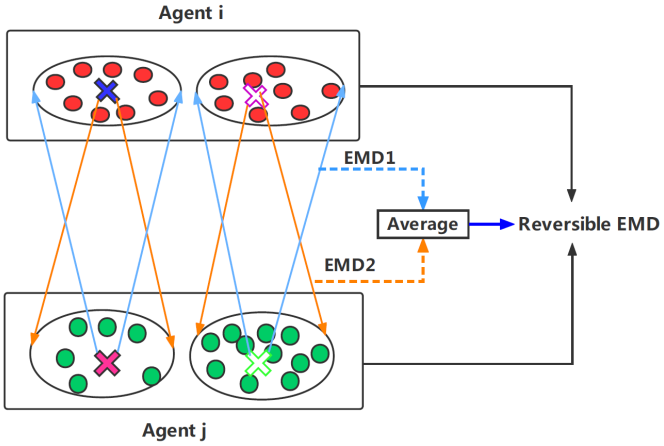

Fig. 4: Reversible EMD between agents

Note that K-means is a stochastic algorithm and it performs bad in some arbitrary shaped datasets. Representing raw data as cluster model could reduce the computation cost significantly, but create a risk for building a less accurate global model. Further more, EMD distance is not symmetric. Given this, we propose a new method reversible EMD. It compares the cluster model of one agent with raw points of another agent, and vice versa, and it is defined as the average distance of these two one-way EMD distances (refer to figure 4). Intuitively, the computation cost is higher than weighted EMD, but it is much accurate.

Compared to the standard EMD method, our proposed method is much more computational efficient. The EMD metrics is applied to the cluster models, rather than raw data, which will substantially reduce the computation cost. Moreover, this measurement could be extended to scenarios that the number of clusters are not consistent over the network, which address the problem of how to choose centroids caused by clustering algorithms themselves.

B. Estimate the number of patterns

Given the similarity measurement, we could infer the number of patterns, which indicates the observed phenomena in the network. We could cluster the agent hierarchically, and select a cut-off value to split the dendrogram into several groups, such that similar agents are assigned in the same group and dissimilar agents are separated into different groups.

The cut-off value, could be estimated by the elbow method, which aims to find the biggest increment in distance during the merging process. Algorithm 1 shows the basic steps. Whenever

there is a new merge, the difference between this merge and last merge is measured (line 3). The merge step, where the difference is maximised, will be marked, and the cut-off value will be narrowed to be the range of distance before merging to distance after merging.

Algorithm 1: Estimate the cut-off value

```
1  Input: Distance metrics D ={d_1, d_2, ··· , d_m};
2  Output: The estimated cut-off value c;
3  Initialised the metric Σ=[];
4  for i = 1; i ≤ m do
5      Σ[i] = d[i+1] - d[i] ;
6  Find the index ι of the maximum value in Σ ;
7  c = D[ι];
```

Example 1. We use a 10-agent network as an example. After each agent compute their provisional models, a distance metrics are used to measure the similarity of agents, and those pair-wise similarities are used to construct a similarity matrix, which is further used as the input for hierarchical clustering. Figure 5 shows the hierarchical clustering result over the EMD matrix.

The distance between the new cluster C_1 (merged by A_6, A_1, A_5, A_9 and A_7) and new cluster C_2 (merged by the other 5 agents) is 4.95, nearly 2 times larger than the last merge (2.05). It indicates that C_1 and C_2 are different and the merge step should stop. Thus, a cut-off value should be selected from the range (2.05, 4.95), and the estimated number of patterns is 2.

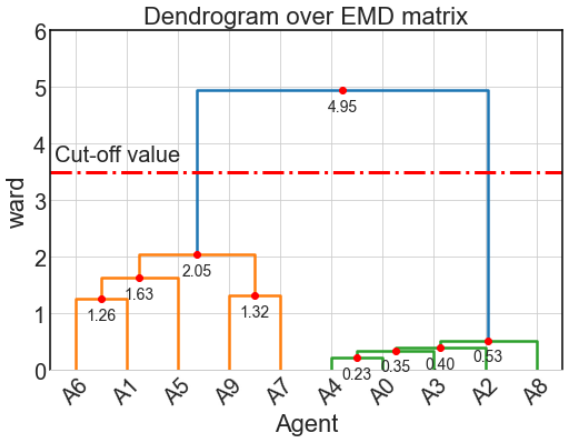

Fig. 5: An example for hierarchical clustering over the EMD matrix

C. Asynchronous Distributed Clustering algorithm

After the sub-groups have been detected, we are trying to learn the global pattern for each sub-group. In this section, we describe the asynchronous distributed clustering algorithm for all agents to learn the global pattern. We assume that agents know the size of network and could end up with different models [1].

[1]More scenarios are considered in our preceding paper [42]. The measure with the best results will be used in other scenarios.

Algorithm 2: Asynchronous distributed clustering algorithm

```
1  Initialization: generate initial centroids and do local
     clustering;
2  Share cluster models with neighbours;
3  while not converged do
4      if received new messages then
5          Update provisional models;
6          Share with neighbours;
7          if received messages from all agents then
8              Compute the global model;
9              Send terminate message to neighbours;
10         else if received terminate message then
11             Accept this model as global model;
12             Inform neighbours of the global model;
13     else
14         Wait until receive a message;
15         if no message is delivered for a specified
             seconds then
16             Converge ;
```

Algorithm 2 shows the basic steps to learn the global model when agents know the size of network. An agent will do repeated updating until the global model is obtained. An agent will compute the global model if it receives messages from all agents in the network, or it is informed the global model (included in the terminate message) from neighbours and accept that global model as its final model. In the extreme case, an agent terminates if there are no incoming messages. Note that different instantiations of this framework could be developed. For instance, the agent could start clustering with a pre-defined $\boldsymbol{k}$ or obtain the initial model with a self-estimated $\boldsymbol{k}$.

How to determine the hyperparameter K is of great importance to most clustering algorithms. We use the Silhouette method to pick the number of clusters. It measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation). The cohesion is measured based on the distance between all the points in the same cluster and the separation is based on the nearest neighbour distance. Formally:

$$Sil(C) = 1/N \sum_{C_k \in C} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{max\{a(x_i, c_k), b(x_i, c_k)\}} \tag{1}$$

where

$$a(x_i, c_k) = \frac{1}{|c_k|} \sum_{x_j \in c_k} d_e(x_i, x_j) \tag{2}$$

$$b(x_i, c_k) = \min_{c_l \in \{C - c_k\}} \{\frac{1}{|c_l|} \sum_{x_j \in c_l} d_e(x_i, x_j)\} \tag{3}$$

The silhouette value ranges from -1 to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to nearest clusters. If most objects have

a high value, then the clustering configuration is appropriate, otherwise, there may be too many or too few clusters.

Algorithm 3: Silhouette method to pick the number of clusters

```
Input: Range of number of clusters: k = {k_0, k_1, ..., k_n};
Output: The number of clusters that fits model best k_b;
for i = 0; i ≤ n do
    Compute clustering algorithm for different values k_i ;
    Calculate the average silhouette (S);
Plot the curve of S for all different values k_i;
Find out the location k_b with maximum S in the plot;
```

Algorithm 3 shows that the silhouette method is similar to the elbow method but the objective is different. Compared to the elbow method, the silhouette method can be automated, since it is simple to determine the maximum value [43]. Further more, Olatz et al. shows that Silhouette achieves the best results among the 30 cluster validity indices in most cases [44]. Note that only the range of the number of clusters has to be fixed in advance. In this paper, we apply the silhouette method to estimate the initial number of clusters $\boldsymbol{k}$ for partition-based clustering method.

IV. Performance Evaluation

In this section, we show the empirical results. First, we show the experimental setting and basic assumptions, followed by the introduction of performance measurement. After that, we consider two different scenarios: the number of clusters is consistent and different.

A. Experimental settings

The four similarity measurements are embedded into the distributed algorithm, and the overall performance of the algorithm is measured. The environmental condition for the clustering algorithm is: 1) we assume a peer-to-peer underlying network with no distinguished agents; 2) agents observe some trigger (i.e., message or a specific command forwarded by neighbours) to initialised the clustering; 3) we assumes that the underlying data could be clustered by various well known algorithms. As a first step, in the paper, we restrict our raw data samples to 2D data. Specifically, three different shaped synthetic datasets, that generated from symmetric Gaussians, asymmetric Gaussians and uniform distributions, are employed to evaluate the performance of proposed algorithm.

An Asynchronous Message Delay Simulator, based on [45], was used to evaluate the performance of our proposed algorithms. Each agent initialises after a random delay (uniformly selected from the range [0,0.3]) and we assume a reliable message delivery, but with some specific delay (random selected from [0.5,1]). The experiments are implemented in a 10-agent network and repeated for 50 runs to eliminate the random oscillation. In the experiments, we consider two different scenarios for the division of agents: 7:3 and 6:4. Taking the first split as an example, it indicates that the first 7 agents are sampling data from one distribution and the rest of 3 agents are sampling from another distribution. The ground truth number of clusters is 5 for each agent. For the experiment with consistent number of clusters, all agents compute the local model with the same $\boldsymbol{k}$. When testing for different $\boldsymbol{k}$, agent estimates $\boldsymbol{k}$ based on its local raw data, so it is possible that some agents or all agents will end up with a different $\boldsymbol{k}$.

B. Performance measurement

In this section, we describe the measurements for evaluating the performance. The convergence time, which measures how fast the algorithms converge, the estimated number of patterns, which measures how correct the estimated patterns are, and clustering accuracy, which measures how correct the raw data point are assigned, are measured in this paper. In addition to accuracy against centralised baseline (denoted by Acc_c), we measure another two accuracies: 1) accuracy against the ground truth (denoted by Acc_g), which compare the final assignment with the ground truth assignment; 2) accuracy against patterns (denoted by Acc_p), which measures the correctness of agent assignment of sub-patterns by comparing to the ground truth sub-patterns [2].

The accuracy measurements used in paper [47] only consider the case of True Positive (TP), where only data points that are assigned to the same cluster as centralised method are taken into account. Here, we consider the data point assignments in a pair-wise way, and the Rand Index (RI) is used to measure the performance. It computes the number of pairs that made the correct decisions by comparing with the benchmark classifications[48]:

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \tag{4}$$

Note that RI assumes that the importance of TP and TN is equal. In some cases, this is not true. Based on that, we consider a weighted version of rand index WRI:

$$WRI = \frac{TP + \delta TN}{TP + FN + \delta FP + \delta TN} \tag{5}$$

where the importance of TN is reduced by a weight δ ($0 \leq \delta \leq 1$). Inspired by this, we also want to balance the contribution of FN by parameter δ, where the recall is weighted. To this end, the F-measure is used as below [49]:

$$F = \frac{(\delta^2 + 1) * P * R}{\delta^2 * P + R} \tag{6}$$

where precision P is defined as $P = \frac{TP}{TP+FP}$ and recall R is denoted by $R = \frac{TP}{TP+FN}$. Note that how agents are split poses a vital effect on the parameter δ, and we assume that δ is defined as the ratio of corresponding size of two sub-patterns.

Notations. We show an simple example to describe all methods in this paper. Consider two agents A_1 and A_2. Each agent is represented by its cluster model, including centroids C_i and counts η_i. Dataset Γ_i is regenerated from the summary description sent from neighbour. Table I shows the a short description of all methods used in the experiment.

[2]More details for this measurement are described in our preceding paper [46].

Name	Descriptions
L1-norm	Measures the L1-norm of euclidean distance between centroids C_1 and C_2.
L2-norm	Measures the L2-norm of euclidean distance between centroids C_1 and C_2 is computed.
Weighted EMD	Computes the EMD between centroids C_1 of agent A_1 and C_2 of agent A_2
Reversible EMD	Computes the average of EMD between A_1's centroids C_1 to A_2's sampled points Γ_2, and the reverse
Standard EMD	Computes the EMD between sampled dataset Γ_1 of agent A_1 and Γ_2 of agent A_2
Wavelet EMD	Computes the Wavelet EMD between regenerated dataset Γ_1 of agent A_1 and Γ_2 of agent A_2
Robust EMD	Computes the Robust EMD between regenerated dataset Γ_1 of agent A_1 and Γ_2 of agent A_2

TABLE I: Description of different methods

C. The number of clusters is consistent

First, we compare the EMD-based method with the widely-used L1- and L2-norm method when the number of clusters is pre-defined as the same. In this experiment, we take the accuracy against ground truth and the number of patterns into account. Figure 6 shows that a higher accuracy against ground truth with a lower standard deviation, is achieved if EMD based methods are used.

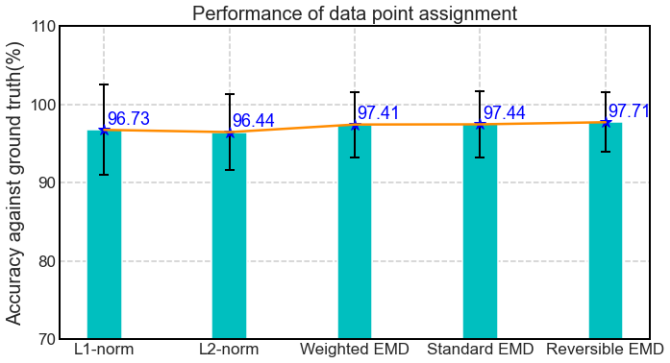

Fig. 6: Accuracy against ground truth with different similarity metrics

Figure 7 shows the estimated number of patterns by these methods. Obviously, algorithms using EMD based methods predict the right number of patterns as the ground truth (with a standard deviation 0). L1- and L2- norm predicts there are around 2.2 patterns in the network with a high standard deviation, which indicates that these two methods are not robust. Our experimental results show that estimated number of patterns for L2-norm could be as high as 8.

Overall, the EMD based method performs well in representing the similarity of agents when the number of clusters are consistent. It achieves a higher clustering accuracy and predicts a more accurate number of patterns. It is worth noting that, in many practical applications, the number of clusters is not necessarily consistent if we accept that agents receive raw data from different phenomena (i.e., different temperature readings reported by sensor in different spaces), where the standard L1- and L2-norm are unable to handle.

D. The number of clusters is different

Then we consider the scenarios where the number of clusters for agents is different. Figure 8 shows the three accuracies

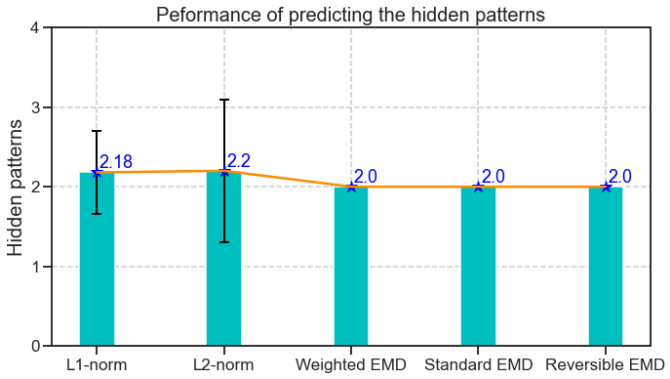

Fig. 7: The predictions of number of patterns with different similarity metrics

when the datasets are generated from asymmetric Gaussian distributions. The top row shows that the accuracy against centralised K-means achieved by these methods. The highest accuracy is obtained by EMD, followed by weighted EMD. Note that the difference between them is negligible (less than 1%). Wavelet EMD performs the worst, but the accuracy is still above 90%. It suggests that all these five methods perform well in this scenario. The row in the middle of Figure 8 shows that all measurement shows a similar trend. Again, there are little differences in these accuracies.

The first two rows shows that there is little difference in measuring the performance of data assignment, but we expect that accuracy against pattern will vary much with these methods. The last row of Figure 8 shows the performance in predicting the group assignment for agents. Weighted EMD achieves the highest accuracy, followed by EMD and reversible EMD. Again, wavelet EMD performs the worst. Overall, the accuracy achieved by weighted EMD, Reversible EMD and EMD are higher than robust EMD (by as much as 16%) and wavelet EMD (by more than 21%). Since the similarity measurement plays a vital role in predicting the group assignment for agents, the method that describes the similarity well is much more likely to make a right prediction. So weighted EMD performs better than other methods.

Figure 9 shows the performance when underlying datasets are generated from partially symmetric Gaussian distributions. Compared to figure 8, the accuracy against centralised K-means has dropped. But still, EMD achieves the highest accuracy, followed by weighted EMD. The second row shows the similar trend and the accuracy remains stable as figure 8, which indicates that accuracy against centralised method is not reliable in some cases. The last row of Figure 9 shows that weighted EMD and EMD outperforms other methods in accuracy against pattern. Again, wavelet EMD achieves the lowest accuracy, which is lower than weighted EMD by as much as 18%. It worth noting that the F-measure method performs better for weighted EMD and EMD. The most possible reason behind this is that similar agents are more likely to be assigned into different sub-patterns with reversible EMD, Wavelet EMD and Robust EMD, and F-measure penalise this wrong prediction.

Figure 10 describes the accuracy when the underlying raw data are generated from uniformly distributed datasets. Simi-

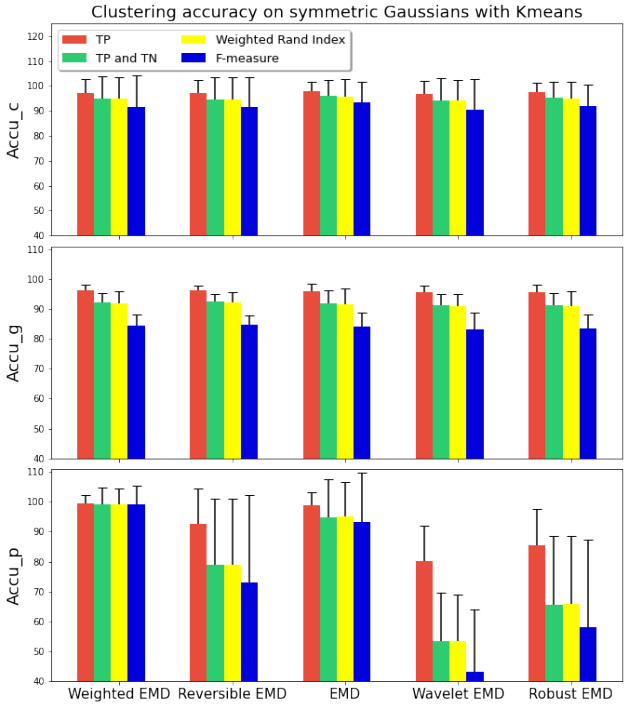

Fig. 8: Performance with symmetric multi-dimensional Gaussians

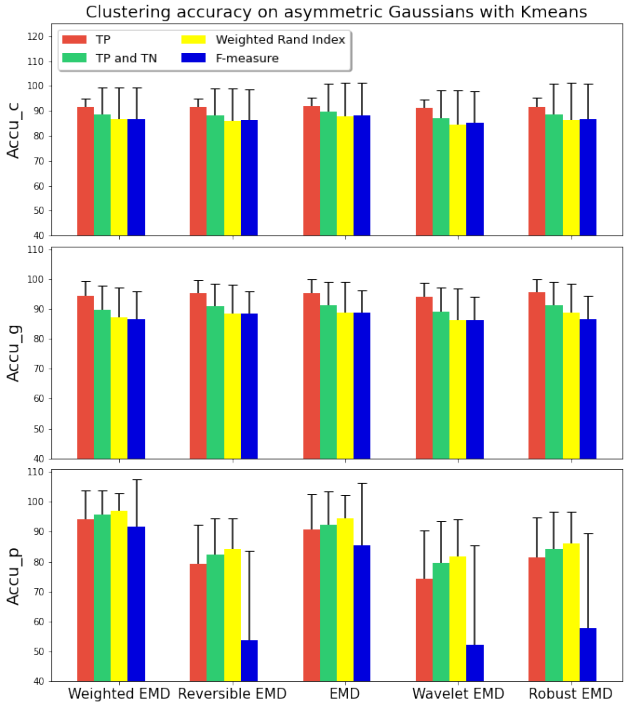

Fig. 9: Performance with asymmetric multi-dimensional Gaussians

larly, there is negligible difference in predicting the assignment for data points (accuracy against centralised method in the first row and accuracy against ground truth in the second row), and the accuracies achieved by these methods are as high as 90%. However, weighted MED outperforms other methods in accuracy against pattern, with an accuracy as high as 92%. Wavelet EMD and robust EMD perform the worst, with an accuracy as low as 75%.

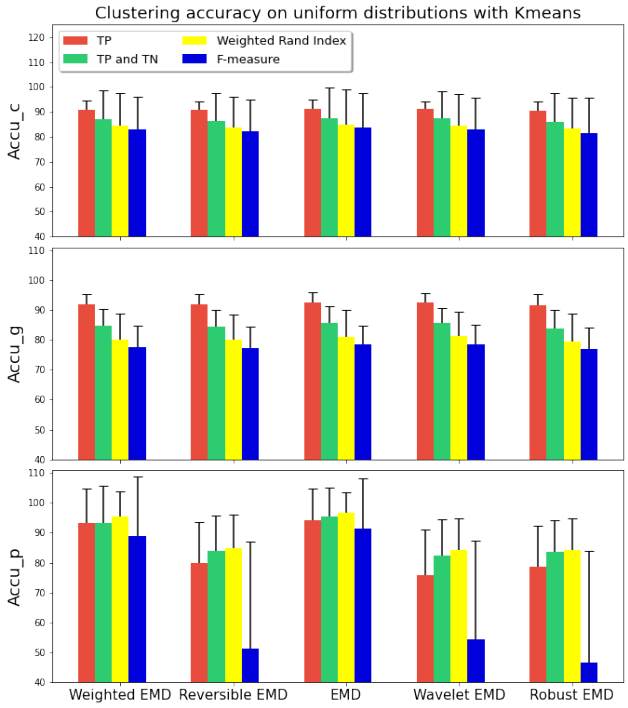

Fig. 10: Performance with uniform distributions

The measurement above shows the performance in predicting data point assignment and estimating the right patterns. However, we have not consider the computation cost yet. We believe that the convergence time, which measures the complete time of whole network, is a good proximity for the computation cost, since it takes a longer time for the network to stabilise if the computation cost is heavy.

Figure 11 shows the convergence time with these five methods over 50 runs. Obviously, wavelet EMD requires the least time to converge, followed by weighted EMD. Note that the gap is negligible since weighted EMD is slightly slower (by less than 1%). Robust EMD converges slower than other methods, since it takes too much time for the set-up step. Although the wavelet EMD converges the fastest, it achieved a lower accuracy against patterns than EMD and weighted EMD (by as much as 20%).

Overall, weighted EMD retains a similar accuracy as EMD in predicting the data point assignment and estimating the right sub-patterns, but converges substantially faster (by nearly 2 times). That is because the raw data are represented as cluster model, and it reduces the computation cost significantly if the work of moving cluster model is considered instead.

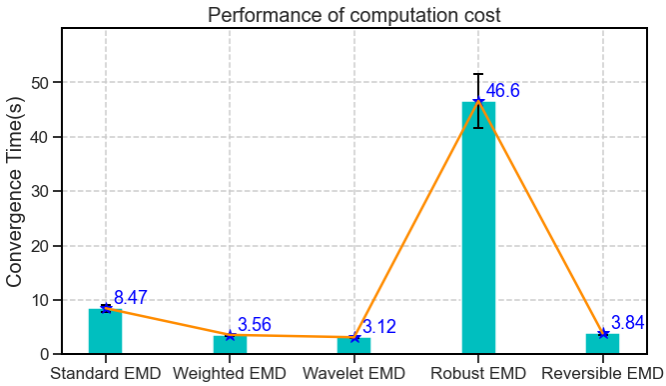

Fig. 11: Comparison of convergence time

E. Evaluation on real-world data

We have obtained the GPS trajectory data from the Transport Commission of Shenzhen Municipality in China, which records the GPS trajectory data of about 20,000 taxis in Shenzhen and 13,000 taxis in Dongguan. The recording devices in taxis report information per minute, which mainly contains aspects: ID of the taxi, time, occupation index, driving speed, direction, latitude, and longitude (see Figure 12).

Given this dataset, we aim to identify whether the exchange of cluster model (i.e., driving speed and location) could improve the driving experience that on the same route. We narrow our experimental areas to Citizens' center, Window of World and University of Shenzhen by restricting the value of latitude and longitude. Figure 13 shows the GPS trajectory of Citizens' center with restricted latitude and longitude ($22.5244 \leq \textit{latitude} \leq 22.5518$ and $114.032 \leq \textit{longitude} \leq 114.0753$).

Based on three experimental areas, we also restrict our time period to one weak. Table II shows the summary of these three experimental areas. For instance, there are around 9,501 taxis picking up and dropping off passengers around the Citizen Centre, and the speed is around 21.29 km/h on average, with a high standard deviation of 23.65. Note that the standard deviation of average speed is highly varied. The possible reason is that the drivers have to drop off passengers frequently when they arrived at the destination, then the driving speed is recorded as 0. Given that, we filter the GPS data and only keep the taxis with complete routes inside this area. In addition, we assume that the driving route is based on the shortest path since knowing the actual route is not an easy task. This shortest path route could be inferred by the software ArcGIS. Figure 14 shows the shortest path route of taxi 195F5 inferred by ArcGIS. One advantage of this method is that it could remove some noisy GPS trajectories (denoted by the green star) caused by the error of the GPS system.

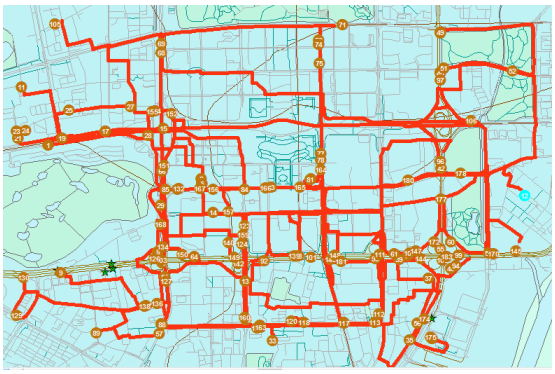

Fig. 14: Trajectory of 195F5

Name	GPS Data N	Taxis M	Attribute	μ	δ
Shenzhen University	148445	6646	AS	38.29	32.731
Citizen Centre	3168850	9501	AS	21.29	23.654
Window of World	202499	7572	AS	27.00	24.465

TABLE II: Overview of Data

The main experimental areas are located on several sequential crossroads. If a warning of traffic jams ahead is altered, a driver could turn left or right to avoid this waiting time. Thus, we measure the driving speed on average and compare it against the ground truth. Note that models from different vehicles may be different, thus it is important to aggregate these different models and compute an accurate model. Table III shows the experimental results on three areas with filtered GPS data. Note that only taxis with more than 200 GPS trajectories within a week are used. Obviously, the speed is improved on average if cluster models of drivers are shared and informed. The traffic collision or red lights could be avoided, thus the driving speed is increased. However, there is only slight improvement around Shenzhen University and Window

粤B341W1,2011-08-31 23:50:49,1,40,315,22.564898,113.888458
粤B341W1,2011-08-31 23:49:29,1,0,315,22.563587,113.889832
粤B052U3,2011-08-31 23:58:46,1,83,0,22.604458,114.117119
粤B49F11,2011-08-31 23:58:46,0,0,315,22.536747,114.107414
粤B43F98,2011-08-31 23:58:46,0,65,90,22.566208,114.111176
粤B0272D,2011-08-31 23:52:16,0,0,180,22.583893,113.949501
粤B537W1,2011-08-31 23:58:42,0,41,135,22.656878,114.017296
粤B64F40,2011-08-31 23:58:44,0,0,90,22.551001,114.06633
粤B3247B,2011-08-31 23:58:44,0,0,0,22.521267,113.931114
粤B09T80,2011-08-31 23:58:44,0,0,90,22.521767,114.031464
粤BYW802,2011-08-31 23:58:27,0,18,135,22.564917,114.039185
粤BA1C94,2011-08-31 23:58:44,1,15,135,22.522783,113.92218
粤B074U1,2011-08-31 23:54:03,0,29,315,22.611082,114.120476
粤B75M61,2011-08-31 23:58:44,1,0,180,22.523968,114.068283
粤B631W0,2011-08-31 23:58:45,1,0,315,22.599421,113.844292
粤B835W2,2011-08-31 23:58:43,1,0,135,22.544718,113.930885
粤B79F15,2011-08-31 23:58:44,0,15,225,22.549183,114.085915
粤B47N71,2011-08-31 23:58:44,0,8,225,22.697416,113.992615

Fig. 12: the Record of GPS data

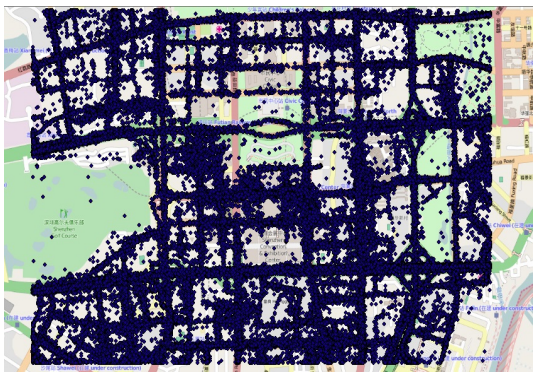

Fig. 13: GPS trajectory around Citizens' center

of World. The most possible reason is that there is heavy traffic congestion inside these two areas. A driver can not do anything even if she knows how to avoid the traffic jams ahead.

Name	GPS Data	Taxis	Attribute	μ	δ
Shenzhen University	9450	46	AS	37.39	12.73
			AS(clustered)	38.29	11.14
Citizen Centre	18550	91	AS	21.29	13.32
			AS(clustered)	25.12	12.43
Window of World	12899	62	AS	27.00	15.46
			AS(clustered)	27.89	13.71

TABLE III: Experimental results with filtered GPS data

To summarise, weighted EMD is an efficient method to measure the similarity between agents in resource constraint networks. It achieves a high accuracy and predicts the right sub-pattern for agents, with a much less computation cost. Experimental results in homogeneous network (where the number of cluster is consistent), show that weighted EMD achieves higher accuracy and predicts the right number of patterns. Experimental results in heterogeneous network (where the number of clusters is different), show that it achieves a similar accuracy as EMD, but much higher than robust EMD and wavelet EMD. Moreover, it converges 2 times faster than EMD.

V. Conclusion and future work

In this paper, we proposed a computational efficient similarity measurement for agents to learn global models, which copes with the salient characteristics of IoV. Specifically, a simplified version of EMD is adopted to measure the similarity between different agents over cluster models, which reduces the computation cost but preserves the instinctive advantages of globally shape-aware and robust to noise. This method works even the resource cross the network is heterogeneous. An Asynchronous Message Delay Simulator is used to evaluate the efficiency of this light-weight method. Empirical results show that weighted EMD achieves similar accuracy as EMD, but it converges significantly faster. Compared to other EMD-based methods, weighted EMD performs better in accuracy than robust EMD and wavelet EMD, requires less time to converge than robust EMD and converges slightly slower than wavelet EMD. Experiments on real-world data collected from vehicles also reinforce the efficiency of this algorithm. To summarise, weighted EMD is a computational efficient method for similarity measurement, which is of great importance for resource constraint Vehicular Network.

In near future, we will consider resource constraint networks with systematical heterogeneity (i.e., network connectivity). We will consider more scenarios where the distributions change over time and different requirements for the network (i.e., all agents are required to end up with the same model) are needed.

Acknowledgment

This research was supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289-P2 and 16/SP/3804, with additional support from National Natural Science Foundation of China under Grant No. U20B2046 and 62202114, China Postdoctoral Science Foundation No. 2022M710861, Guangdong Basic and Applied Basic Research Foundation No. 2020A1515010450 and No. 202102020867, Basic Research Program Co-funded by Guangzhou City and Guangzhou University No. 202102010445, Joint Research Fund of Guangzhou and University under Grant No. 202201020380, and Guangdong Higher Education Innovation Group No. 2020KCXTD007.

References

[1] V. E. F. C. Borges, D. F. S. Santos, A. Perkusich, and K. M. Malarski, "Survey and evaluation of internet of vehicles connectivity challenges," in *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2020, pp. 1–6.

[2] K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier, "Cooperative intelligent transport systems in europe: Current deployment status and outlook," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 89–97, 2017.

[3] H. Zhu, S. Chang, W. Zhang, F. Wu, and L. Lu, "Online vehicle front–rear distance estimation with urban context-aware trajectories," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1063–1074, 2018.

[4] M. Alam, J. Ferreira, and J. Fonseca, *Introduction to Intelligent Transportation Systems*. Cham: Springer International Publishing, 2016, pp. 1–17.

[5] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," *IEEE International Conference on Data Engineering*, 2021.

[6] Z. Chen, P. Tian, W. Liao, and W. Yu, "Zero knowledge clustering based adversarial mitigation in heterogeneous federated learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1070–1083, 2021.

[7] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3034–3048, 2020.

[8] W. Huang, M. Ye, and B. Du, "Learn from others and be yourself in heterogeneous federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 10 143–10 153.

[9] M. Pabian, D. Rzepka, and M. Pawlak, "Supervised training of siamese spiking neural networks with earth mover's distance," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4233–4237.

[10] H. Phan and A. Nguyen, "Deepface-emd: Re-ranking using patch-wise earth mover's distance improves out-of-distribution face identification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 20 259–20 269.

[11] P. Liang, Y. Zhang, Y. Ding, J. Chen, C. S. Madukoma, T. Weninger, J. D. Shrout, and D. Z. Chen, "H-emd: A hierarchical earth mover's distance method for instance

segmentation," *IEEE Transactions on Medical Imaging*, 2022.

[12] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," ser. NIPS'13. Red Hook, NY, USA: Curran Associates Inc., 2013, p. 2292–2300.

[13] J. Altschuler, J. Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via sinkhorn iteration," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 1961–1971.

[14] G. Luise, A. Rudi, M. Pontil, and C. Ciliberto, "Differential properties of sinkhorn approximation for learning with wasserstein distance," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, 2018, p. 5864–5874.

[15] F. Bai, D. D. Stancil, and H. Krishnan, "Toward understanding characteristics of dedicated short range communications (dsrc) from a perspective of vehicular network engineers," in *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '10, 2010, p. 329–340.

[16] L. Cagliero, M. La Quatra, and D. Apiletti, "From hotel reviews to city similarities: A unified latent-space model," *Electronics*, vol. 9, no. 1, 2020.

[17] O. Nežerenko, O. Koppel, and T. Tuisk, "Cluster approach in organization of transportation in the baltic sea region," *Transport*, vol. 32, no. 2, pp. 167–179, 2017.

[18] R. de Oña, G. López, F. J. D. de los Rios, and J. de Oña, "Cluster analysis for diminishing heterogeneous opinions of service quality public transport passengers," *Procedia - Social and Behavioral Sciences*, vol. 162, pp. 459–466, 2014.

[19] T. Pei, W. Wang, H. Zhang, T. Ma, Y. Du, and C. Zhou, "Density-based clustering for data containing two types of points," *International Journal of Geographical Information Science*, vol. 29, no. 2, pp. 175–193, 2015.

[20] Z. JIANG, S. LI, and X. LIU, "Parameters calibration of traffic simulation model based on data mining," *Journal of Transportation Systems Engineering and Information Technology*, vol. 12, no. 6, pp. 28–33, 2012.

[21] M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, "Data mining and machine learning methods for sustainable smart cities traffic classification: A survey," *Sustainable Cities and Society*, vol. 60, p. 102177, 2020.

[22] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[23] S. Chang, Y. Qi, H. Zhu, J. Zhao, and X. Shen, "Footprint: Detecting sybil attacks in urban vehicular networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, pp. 1103–1114, 2012.

[24] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 779–811, 2019.

[25] O. Araque, G. Zhu, and C. A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowledge-Based Systems*, vol. 165, pp. 346–359, 2019.

[26] N. Saeed, H. Nam, M. I. U. Haq, and D. B. M. Saqib, "A survey on multidimensional scaling," *ACM Computing Surveys (CSUR)*, vol. 51, pp. 1–25, 2018.

[27] A. Agarwal, M. Chauhan, and Ghaziabad, "Similarity measures used in recommender systems : A study," 2017.

[28] M. R. Deac-Petruşel, "A comparative analysis of similarity measures in memory-based collaborative filtering," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2020, pp. 140–151.

[29] R. Zhang, F. Nie, Y. Wang, and X. Li, "Unsupervised feature selection via adaptive multimeasure fusion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2886–2892, 2019.

[30] C. R. Wren, D. C. Minnen, and S. G. Rao, "Similarity-based analysis for large networks of ultra-low resolution sensors," *Pattern Recognition*, vol. 39, no. 10, pp. 1918–1931, 2006.

[31] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Model. Meth. Appl. Sci.*, vol. 1, 01 2007.

[32] H. Sun, Y. Peng, J. Chen, C. Liu, and Y. Sun, "A new similarity measure based on adjusted euclidean distance for memory-based collaborative filtering," *J. Softw.*, vol. 6, pp. 993–1000, 2011.

[33] Y. Tang, L. H. U, Y. Cai, N. Mamoulis, and R. Cheng, "Earth mover's distance based similarity search at scale," *Proc. VLDB Endow.*, vol. 7, no. 4, p. 313–324, dec 2013.

[34] S. Shirdhonkar and D. W. Jacobs, "Approximate earth mover's distance in linear time," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[35] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," in *Computer Vision – ECCV*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin Heidelberg, 2008, pp. 495–508.

[36] H. Yuan and T. Ma, "Federated accelerated stochastic gradient descent," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5332–5344, 2020.

[37] S. Ahn, J. Kim, E. Lim, W. Choi, A. Mohaisen, and S. Kang, "Shmcaffe: A distributed deep learning platform with shared memory buffer for hpc architecture," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, 2018, pp. 1118–1128.

[38] M. S. Hadi, A. Q. Lawey, T. E. El-Gorashi, and J. M. Elmirghani, "Big data analytics for wireless and wired network design: A survey," *Computer Networks*, vol. 132, pp. 180–199, 2018.

[39] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino, "Fault tolerant decentralised k-means clustering for asynchronous large-scale networks," *Journal of Parallel and Distributed Computing*, vol. 73, no. 3, pp. 317–329, 2013.

[40] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," in *2010 IEEE International*

Symposium on Information Theory, 2010, pp. 1753–1757.

[41] M. Bendechache, N.-A. Le-Khac, and M.-T. Kechadi, "Hierarchical aggregation approach for distributed clustering of spatial datasets," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2016, pp. 1098–1103.

[42] C. Qiao, K. N. Brown, F. Zhang, and Z. Tian, "Federated adaptive asynchronous clustering algorithm for wireless mesh networks," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[43] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[44] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. PéRez, and I. Perona, "An extensive comparative study of cluster validity indices," *Pattern Recognition*, vol. 46, no. 1, pp. 243–256, 2013.

[45] Y. Mechqrane, M. Wahbi, C. Bessiere, and K. N. Brown, "Reordering all agents in asynchronous backtracking for distributed constraint satisfaction problems," *Artificial Intelligence*, vol. 278, p. 103169, 2020.

[46] C. Qiao, J. Qiu, Z. Tan, G. Min, A. Y. Zomaya, and Z. Tian, "Evaluation mechanism for decentralized collaborative pattern learning in heterogeneous vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2022.

[47] S. Datta, C. Giannella, and H. Kargupta, "Approximate distributed k-means clustering over a peer-to-peer network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1372–1388, 2008.

[48] Y. Chen and J. W. Baker, "Community detection in spatial correlation graphs: Application to non-stationary ground motion modeling," *Computers & Geosciences*, vol. 154, p. 104779, 2021.

[49] D. Hand and P. Christen, "A note on using the f-measure for evaluating record linkage algorithms," *Statistics and Computing*, vol. 28, no. 3, pp. 539–547, 2018.

Cheng Qiao Cheng Qiao is a Postdoc researcher in Guangzhou University. He received the Ph.D degree from University College Cork, Ireland, in 2021. He completed his M.S. from Fujian Normal University, China in 2013 and B.E from Xidian University, China in 2010. Before joining UCC, he worked as a Research Assistant in the Shenzhen Institute of Advanced Technology, Chinese Academy of Science (CAS). His research interests include clustering algorithms, inference learning, intelligent decision-making and distributed learning in Wireless Networks.

Kenneth N. Brown Kenneth N. Brown is a Professor of Computer Science at University College Cork in Ireland. He joined UCC Computer Science Department as a senior lecturer in 2003. Prior to that he was a lecturer at the University of Aberdeen, a Research Fellow at Carnegie Mellon University, and a Research Associate at the University of Bristol. His research interests are in the application of AI, optimisation and distributed reasoning, with a particular focus on wireless networks. He is a Co-Principal Investigator/Group Leader in Insight: Centre for Data Analytics.

Yong Zhang Yong Zhang is currently a Professor in Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Science (CAS), Honorary Professor in the University of Hong Kong. He received his Ph.D In the Department of Computer Science and Engineering at Fudan University in 2007. Before joining SIAT, he worked as Post-Doctoral Fellow and Senior Researcher in TU-Berlin and HKU, respectively. His research interests include design and analysis of algorithms, combinatorial optimization and wireless networks. He has published more than 100 papers in refereed journals and conferences.

Zhihong Tian Zhihong Tian is currently a Professor, and Dean, with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangdong Province, China. Guangdong Province Universities and Colleges Pearl River Scholar (Distinguished Professor). He is also a part-time Professor at Carlton University, Ottawa, Canada. Previously, he served in different academic and administrative positions at the Harbin Institute of Technology. He has authored over 200 journal and conference papers in these areas. His research interests include computer networks and cyberspace security. His research has been supported in part by the National Natural Science Foundation of China, National Key Research and Development Plan of China, National High tech RD Program of China (863 Program), and National Basic Research Program of China (973 Program). He also served as a member, Chair, and General Chair of a number of international conferences. He is a Senior Member of the China Computer Federation, and a Senior Member of IEEE.