A secure data sharing scheme in Community Segmented Vehicular Social Networks for 6G

***Abstract*—The use of aerial base stations, AI cloud, and satellite storage can help manage location, traffic, and specific application-based services for vehicular social networks. However, sharing of such data makes the vehicular network vulnerable to data and privacy leakage. In this regard, this paper proposes an efficient and secure data sharing scheme using community segmentation and a blockchain-based framework for vehicular social networks. The proposed work considers similarity matrices that employ the dynamics of structural similarity, modularity matrix, and data compatibility. These similarity matrices are then passed through stacked autoencoders that are trained to extract encoded embedding. A density-based clustering approach is then employed to find the community segments from the information distances between the encoded embeddings. A blockchain network based on the Hyperledger Fabric platform is also adopted to ensure data sharing security. Extensive experiments have been carried out to evaluate the proposed data-sharing framework in terms of the sum of squared error, sharing degree, time cost, computational complexity, throughput, and CPU utilization for proving its efficacy and applicability. The results show that the CSB framework achieves a higher degree of SD, lower computational complexity, and higher throughput.**

***Index Terms*—5G, Internet of Things, Vehicular social networks, Secure Data Sharing, Community Segmentation, Blockchain.**

I. Introduction

THE popularity of vehicular social networks (VSNs) has been increasing recently due to the advancement of wireless communication networks and ubiquitous devices. Considering the scalability of ubiquitous devices, it is not efficient to rely on a single cloud or satellite server to meet quality of service (QoS) demands. In this regard, unmanned aerial vehicles (UAVs) are considered for providing services to roadside vehicles. Connection to a nearby UAV would be much more convenient and spontaneous compared to the cloud or satellite server. For example, vehicles can transmit their information related to traffic events, current location, acceleration, speed, and so forth, to the connected UAV that can be shared with other UAVs for probable danger or accidental situations [1]. Furthermore, the cloud or satellite server can provide optimal traffic analysis by collecting information from multiple UAVs in soft real-time [2].

With the emergence of VSNs, vehicular networks have benefitted a lot in terms of service availability and convenience. However, concerns related to privacy and security have risen rapidly. For example, data shared by an individual vehicle, such as location, can be intercepted by an attacker that can lead to the user's work or home address [3]. Furthermore, manipulation or alteration of information regarding acceleration, speed, or traffic events can affect the judgment of other vehicles that can lead to serious accidents [4]. It is also well-known to VSN community that the devices associated are computational and battery constrained. Moreover, the vehicles move at high speeds that raises the issues of connectivity with the base station. The main reason for considering UAVs as aerial base station in this study is to overcome the connectivity and availability issues concerning base stations. However, UAVs also have limited capacity; therefore, designing a scalable and privacy-preserving scheme for VSNs is still an open issue in the VSN community [4].

The problem of VSNs is similar to that of complex communication networks, where nodes grow at an exponential rate; therefore, division of nodes in different communities is one of the potential solutions to make the beyond 5G communication system scalable. Generally, the nodes that share common information or possess common characteristics can be grouped to form a single community. To the best of our knowledge, the community segmentation method using the deep learning method for VSNs has not yet been exploited. It is well-known that deep learning methods increase the computation complexity of the system which can be a hindrance in meeting flexible deadlines. Researchers are considering reducing the dimensions of feature representation in an efficient manner to reduce this computational complexity while retaining a reasonable accuracy of community detection. In this regard, this study opts for stacked autoencoders as they extract feature embedding while reducing the dimensional space, thus, reducing the computational complexity of the system.

Recently, blockchain has been gaining a lot of interest due to being the underlying technology related to bitcoin as well as yielding the characteristics of immutability and scalability. Blockchain can be considered as a distributed record keeping system that demonstrates the dynamics of decentralization, privacy preservation, and non-tamperability that help to maintain the privacy and security of shared data [5]. Blockchain has been applied to a wide variety of application domains such as power industry, internet of things (IoT), and medical care, while achieving reliable results [6]. There are several blockchain platforms or frameworks available for implementation; however, Ethereum, R3 Corda and Hyperledger Fabric are considered the most preferable platforms by both research and industrial communities[1]. In this study, we opt for Hyperledger Fabric platform which is a permissioned blockchain platform that was developed by the Linux Foundation. The hyperledger platform is selected because of its need-to-know characteristics in data sharing schemes, less storage requirements, transaction speed, active user community, pluggable consensus schemes, modular architecture, quality control, and support for open collaboration.

[1]https://searchcio.techtarget.com/feature/Top-9-blockchain-platforms-to-consider

Although existing studies use blockchain technology to solve data privacy and security issues while presenting the frameworks for privacy protection and data sharing, the details of what type of data needs to be shared along with the desired attributes in the field of VSNs has yet to be explored. Blockchain technology considers a block number, address number, or transaction number to complete a transaction query that is not sufficient to ensure the security of shared data, so it is considered a challenging task [7]. Some existing studies raised the issues regarding the real-time services of VSNs when used in conjunction with the blockchain technology, however, the recent studies have proved that real-time services are achievable while using blockchain and other emerging technologies, i.e. AI, in VSNs, respectively [8], [9]

This paper proposes a community segmentation and blockchain (CSB) framework for an efficient and secure data sharing scheme in VSNs. The data sharing framework in this study is based on Hyperledger Fabric, which acquires huge amounts of vehicle data and transmits them to the blockchain network. Additionally, the community detection algorithm computes the similarity, correlation, and compatibility of the acquired data to segment a large number of vehicles with the intention of sharing VSN data within the matching cluster, presuming that segmentation will help solve the scalability issue. We propose the use of stacked autoencoders for community segmentation algorithm to group the vehicles in an efficient manner while evaluating the efficacy of VSN community division through sharing degree criterion. The main contributions of this study are listed below.

- A CSB framework is proposed for a secure, scalable, and efficient data sharing scheme in VSN.
- Stacked autoencoder based community segmentation method for grouping the vehicles using novel similarity features.
- A Blockchain network workflow based on the Hyperledger Fabric platform to secure the shared data.

The remainder of the paper is structured as follows. Section 2 consolidates the existing works. Section 3 presents the CSB framework for the data sharing scheme and provides details for the community segmentation method. Section 4 presents the experimental results and Section 5 concludes the study together with future work.

II. RELATED WORK

Although many studies have focused on preserving privacy and security for the vehicular network, [10] we consolidate a review of the state-of-the-art work on vehicular social networks accordingly. Some studies use pseudonym strategies or cryptographic methods to preserve vehicle privacy. For instance, Abdallah and Shen [11] proposed the use of authentication strategies that have lightweight characteristics to preserve characteristics for vehicle-to-grid (V2G) connections. Hussain et al. [12] proposed incentives-based vehicle witnesses as a service framework (IVWaaS) that acquires image data from nearby vehicles and roadside cameras to capture the site of interest. The image data is then transferred to the cloud anonymously for preserving privacy. Zhao et al. [13] used a community similarity-based cache policy for vehicular networks. The privacy is preserved by selecting similar vehicles based on community likeness. Kang et al. [1] proposed a pseudonym system that preserves privacy for vehicle networks based on fog computing. The scheme used edge resources with road infrastructure to improve the security of communications carried out between vehicles. Pu et al. [4] proposed the security mechanism for data sharing schemes based on secret sharing systems and attribute-based encryption techniques based on ciphertext policy. Their system was designed specifically for edge servers and economic denial-of-sustainability attacks. Ma et al. [14] proposed the use of attribute-based encryption to secure the announcements shared between vehicular networks. The study considered vehicular attributes to share the data, accordingly.

Researchers have also tried to integrate the characteristics of blockchain into vehicular networks to deal with data security and privacy issues, accordingly. Kang et al. [6] proposed a blockchain-based solution for data sharing security on the Internet of Vehicles (IoV). The study presented a two-stage solution; the first performs a computation on the reputation of a miner, and the second audits the concerned blocks to avoid collision between active and standby miners. The bottleneck, due to the lack of interaction between standby and active miners, was solved using their proposed contract theory. Yao et al. [15] proposed the use of blockchain-assisted authentication mechanisms for vehicular networks to maintain anonymity and protect privacy, respectively. Extensive security analysis was carried out for validating the practicality and efficiency of their proposed method. Su et al. [16] proposed the use of a permissioned blockchain system to ensure secure charging of electric vehicles through smart contracts. The study also used Byzantine fault tolerance as a consensus algorithm in permissioned blockchain systems. The study showed that the blockchain system not only achieves a satisfactory level of security but also enhances energy efficiency. Pu et al. [4] proposed the use of blockchain for preserving security and privacy in data sharing methods with respect to VSNs. The study also used game theory to impose rewards and punishments to protect vehicle information. Jiang et al. [17] also proposed the use of blockchain together with an identity-based authentication mechanism to establish trust among vehicular nodes.

The use of community detection has also been extensively studied in existing studies. Sammarco et al. [18] proposed the use of community segmentation for wireless traffic applications by computing the similarities of the traces. Xia et al. [19] proposed the use of betweenness and closeness to calculate the similarity between social dimensions. Raj et al. [20] proposed a granulation-based factoring method to segment the community for social networks. Liu et al. [21] proposed the use of colony optimization and contiguity-constraints to form spatial vehicle communities to predict their trajectories. Compared to the aforementioned works, the proposed method aims to achieve two objectives. The first is the segmentation of vehicle communities that would improve the energy efficiency and scalability of vehicular social networks. We propose novel similarity matrices and

stacked auto-encoders for segmentation. To the best of our knowledge, similarity matrices that would assume the data compatibility in combination with the stacked auto-encoders for segmentation have not been proposed before. The second is the security and privacy issue concerning the data-sharing mechanism in VSNs that is achieved using Hyperledger fabric platform.

III. PROPOSED METHODOLOGY

The transactions for data sharing services in VSNs concerning beyond 5G networks are classified according to their level of privacy. The blockchain databases characterize these levels based on the availability of data in public domain, i.e. encrypted, community-specific, and public data, respectively. Our assumption in this work considers that the privacy level for the shared data should be set to the community-specific so that the data is available to the vehicles categorized to be in the same cluster. Our work will mostly revolve around the design of a community segmentation method for efficient accessing and sharing of community-specific data. The proposed CSB framework for the data sharing scheme is depicted in Figure 1. The CSB framework comprise three layers, i.e. data-, segmentation-, and blockchain- layer, respectively. The first layer is responsible for data collection from UAVs and vehicles and its transmission to the segmentation layer. The segmentation layer narrows the scope of data sharing by segmenting vehicles into communities. The blockchain layer then secures the transaction records and community segmentation results. The details for each of the layers are given in the subsections below.

A. Data Layer

This layer in CSB framework is mostly concerned mainly with the data generated from vehicles that include sensors, onboard units, machine running state, quantity, product type, and other parameters. The data from these vehicles are collected by the aerial base stations (UAVs). The reasons for considering UAVs for this study are threefold. The first concerns beyond 5G networks, which cater to diverse and stringent requirements through portable base stations that include UAVs [22]. The second is the wide range of use cases that could be handled by the use of UAVs such as remote constructions, real-time surveillance, media production, and package delivery. The third is the increasing demand for UAV deployment, which can reach more than 1.5 million by 2024 according to the Federal Aviation Administration [23]. Furthermore, the concept of unmanned aircraft systems for traffic management (UASTM) is proposed for beyond 5G networks that could operate at low-altitudes to provide the service in high traffic density and low service coverage areas. The data collected by the UAVs can then be used for comprehensive analysis sent to the segmentation layer for a secure data sharing scheme.

B. Segmentation Layer

The segmentation layer comprises vehicles, UAVs, and AI cloud. The UAVs act as an intermediary that collect the data from vehicles and transmits the data to AI cloud and blockchain network. The AI cloud is responsible for segmenting vehicles in communities via our proposed community segmentation method, generating the segmentation results, and updating the blockchain network with the segmented communities, accordingly. Once vehicles have been successfully segmented, they can query the shared data with the help of smart contract execution. One of the unique points of CSB framework is the consideration of data compatibility in the similarity matrix, which is neglected in existing studies. We assume that the reasons community segmentation in such studies does not achieve suitable results is due to the machine learning model or the data shared is incompatible. It is of utmost importance that the shared data are compatible within the community to achieve efficient results and ease of processes. We also hypothetically presume that it could reduce the computation time in the AI cloud, since machine learning models with similar types of data are trained and updated faster compared to heterogeneous data modalities. Another component to highlight is the stacked autoencoder that further helps reduce the computational process by reducing the dimensions of the similarity matrix for community segmentation.

1) Community Segmentation Method: Our proposed community segmentation method is based on 4 steps, i.e., similarity matrix construction, stacked autoencoders, information distance and density-based clustering. In this study, we consider the use of social networks to aggregate all vehicles into communities. The basic theory of social networks is that the relationship among vehicles of the same community would be stronger in comparison to those vehicles belonging to a different community. Our proposed method uses the combination of graph embedding, information theory, and clustering method, for segmenting vehicles into different communities. We represent the topological structure of the network as a graph $Gr = (V_t, E_d)$, where E_d represents the edges or the connection of two nodes and V_t refers to the N number of nodes. The adjacency matrix of Gr is given by a positive symmetric matrix $Adj = a_{pq} \in \mathbb{R}^{N \times N}$ comprising of binary values 0,1, accordingly. For instance, a_{pq} is set to 0 if no egde exists between vt_p and vt_q, and 1, otherwise.

The adjacency matrix intrinsically describes the relationship between the nodes in a network; however, the similarity of the nodes in vehicular social networks is more complicated. Therefore, indirect relationships should also be taken into account when computing the similarity matrices. In this work, we describe three varying functions to measure the similarity between vehicular nodes. The first is the one that is widely used for social networks, that is, the structural similarity matrix, defined in Equation 1.

$$Sim_{str}(p,q) = \frac{|vt(p) \cap vt(q)|}{\sqrt{|vt(p) \times vt(q)|}} \qquad (1)$$
$$vt(p) = \{q \in Vt | (p,q) \in Ed\} \cap \{p\}$$

The numerator describes the number of nodes that are connected to p and q both and the denominator refers to the number of nodes that p and q can be connected with. Leveraging

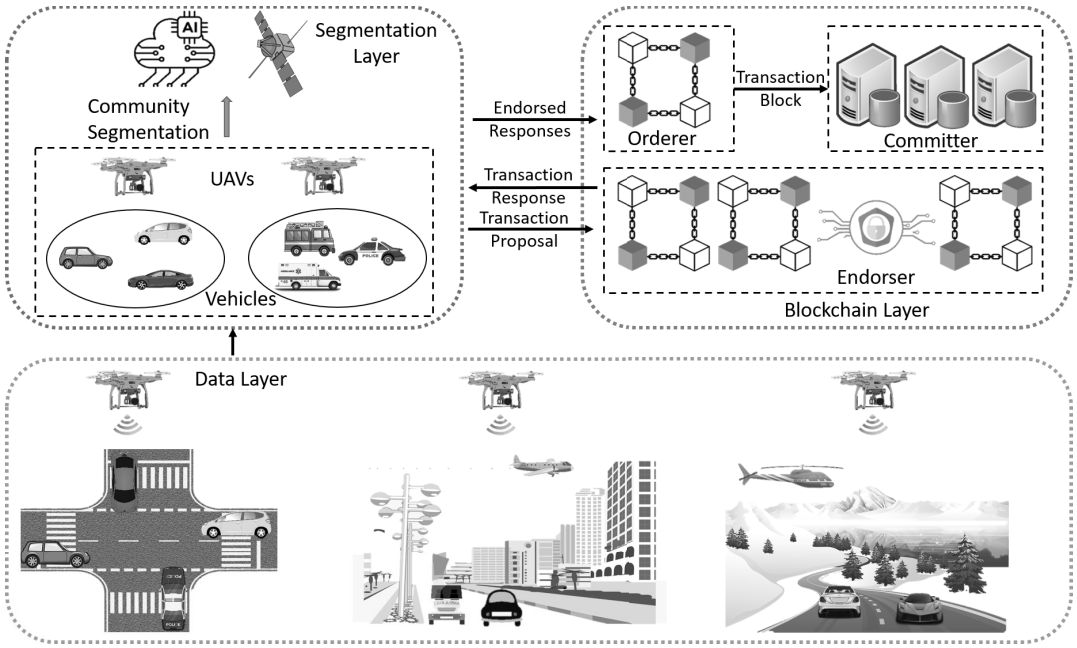

Fig. 1. The proposed community segmentation and blockchain framework for data sharing scheme in vehicular social networks

the structural similarity $Sim_{str}(p,q)$, we compute the distance matrix as shown in equation 2.

$$Dist_{gr} = [d_{pq}]_{N\times N}, d_{pq} = \begin{cases} \frac{1}{Sim_{str}(p,q)}, & \text{if } p \neq q \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The second similarity matrix we consider is the modularity matrix [24] which represents the difference between the expected number of edges that characterize all nodes and the number of edges within the communities. The similarity matrix is shown in Equation 3.

$$Sim_{mod} = [md_{pq}]_{N\times N}, md_{pq} = a_{pq} - \frac{\gamma_p\gamma_q}{2\mu} \quad (3)$$

The term $\frac{\gamma_p\gamma_q}{2\mu}$ refers to the number of edges randomly placed between nodes p and q, μ is the average number of network edges, γ_p and γ_q represent the vertex degree for nodes p and q, respectively. The third similarity matrix is based on the data compatibility. This measure computes the similarity between the feature spaces transformed with the principal component analysis (PCA). PCA has been mostly used for feature extraction or projecting data onto lower dimensions by choosing the coefficients based on their variance. Let the data acquired from a vehicular node and the embedding dimension be represented as x and $\Re$, respectively. The matrix can then be written as shown in Equation 4.

$$x_{pca} = \frac{1}{\sqrt{\iota}} \begin{bmatrix} x_1 & \cdots & x_\iota \\ \vdots & \ddots & \vdots \\ x_\Re & \cdots & x_Z \end{bmatrix}^T = \frac{1}{\sqrt{\iota}} \begin{bmatrix} x_1 \\ \vdots \\ x_\iota \end{bmatrix} \quad (4)$$

where Z refers to the number of samples and $\iota = Z-(\Re-1)$. A covariance matrix $\Lambda \in \mathbb{R}^{\iota\times\iota}$ is computed from x_{pca} as shown in Equation 5.

$$\Lambda_{\iota\times\iota} = \frac{1}{\iota} x_{pca} \cdot x_{pca}^T \quad (5)$$

Once the covariance matrix is obtained, the Eigenvalues λ are calculated and sorted in the descending order. The data is then projected onto lower dimensions using PCA, followed by the computation of Euclidean distance from the feature spaces to construct a similarity matrix as shown in equation 6.

$$Sim_{feat} = [ft_{pq}]_{N\times N}, ft_{pq} = \begin{cases} 1, & \text{if } dist_{euc} < \tau, \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where τ represents the threshold that varies from [0 1] and $dist_{euc}$ refers to the Euclidean distance, accordingly. We assume that a single adjacency matrix representing similarity would lose significant relationship information in a complex network; therefore, multiple similarity matrices are required to map out the node relationship in an efficient manner.

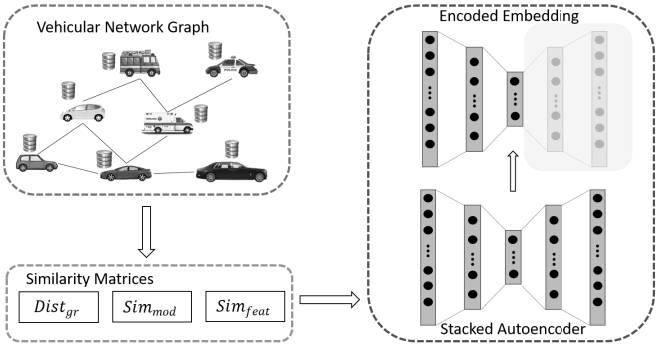

Fig. 2. Encoded Embedding from similarity matrices using stacked autoencoders

2) Stacked Autoencoders: We use the similarity matrices to train the stacked autoencoders that provide latent space representation as shown in Figure 2. The representation of lower-dimensional data in feature space where similar points

are closer, while dissimilar points are farther is suggested to be the latent space. The stacked autoencoders are comprised of input, encoding, decoding, and output layers, accordingly. In this study, the stacked autoencoders are first trained to encode lower-dimensional representations and decode the same matrices from the encoded embedding. Once, the training is complete, the second step is to use the encoded embedding for computing information distances for community segmentation, accordingly. As stacked autoencoders reduce the overall dimensionality of the feature space, it is intuitive to assume that the computational complexity is reduced in comparison to existing studies that either use the full feature space or extract features that are larger in dimensions. The hidden layer representation for encoding part is shown in Equation 7

$$Enc_N = Act(W \cdot S_N + b) \tag{7}$$

where $S = [Dist_{gr}, Sim_{mod}, Sim_{feat}]$, and $Act(\cdot)$ is the activation function of the sigmoid. The terms W and b refer to the weights and biases for the encoder. The decoder part uses the encoded embedding for reconstructing $\hat{S}$, which can be formulated as shown in equation 8.

$$\hat{S}_N = Act(\hat{W} \cdot Enc_N + \hat{b}) \tag{8}$$

where $\hat{W}$ and $\hat{b}$ are the corresponding weights and biases for the decoder. The embedding for the encoder and reconstruction for the decoder can be optimized by minimizing the objective function shown in Equation 9.

$$\begin{aligned} J(\vartheta) &= min_{\vartheta} \sum_{t=1}^{N} \mathcal{L}(S_t, \hat{S}_t) + \Psi \sum_{m=1}^{M} KL(\rho||\hat{\rho}_m) \\ &= min_{\vartheta} \sum_{t=1}^{N} \mathcal{L}(S_t, \hat{S}_t) \\ &+ \Psi \sum_{m=1}^{M} \rho log\frac{\rho}{\hat{\rho}_m} + (log\frac{1-\rho}{1-\hat{\rho}_m})(1-\rho) \end{aligned} \tag{9}$$

The term Ψ corresponds to the parameter that controls the sparsity penalty term. The reconstruction loss between the input and the decoded output is characterized by $\mathcal{L}(\cdot)$, which is defined as $\sum_{t=1}^{N}(\hat{S}_t log(S_t) + (1-\hat{S}_t)log(1-S_t))$. The notation M in the penalty term refers to the number of neurons in the hidden layer. The average activation function is defined by the term $\hat{\rho}_m$. The sparsity penalty term is usually set very close to 0 which is the case in this study. The parameters are optimized iteratively until a minimum reconstruction loss is obtained. The optimization of parameters is performed by the following set of equations shown in Equation 10.

$$\begin{aligned} &W \leftarrow W - \eta\frac{\partial J(\vartheta)}{\partial W}, \hat{W} \leftarrow \hat{W} - \eta\frac{\partial J(\vartheta)}{\partial \hat{W}}, \\ &b \leftarrow b - \eta\frac{\partial J(\vartheta)}{\partial b}, \hat{b} \leftarrow \hat{b} - \eta\frac{\partial J(\vartheta)}{\partial \hat{b}} \end{aligned} \tag{10}$$

The notation η corresponds to the learning rate for updating the hyperparameters. The process of extracting encoded embedding using a stacked autoencoder from the similarity matrices is depicted in figure 2.

3) Information Distance: The output of the stacked autoencoders provides us with $\mathcal{K}$ encoded vectors having lower dimensions. Each encoded vector is assumed to represent a probability distribution with unknown parameters. Therefore, we obtain a series of probability distributions, each representing the embedding of a single vehicular node, $PD = \varphi_1, \cdots, \varphi_{\mathcal{K}}$, which contains the low-dimensional representation of similarity matrices. We then use Kullback-Leibler (KL) divergence to compute the information distance between the probability distributions as shown in Equation 11.

$$dist(\varphi_1, \varphi_2) = \sum \varphi_1 log\frac{\varphi_1}{\varphi_2}, s.t. \varphi \geq 0, and \sum \varphi_{\mathcal{K}} = 1 \tag{11}$$

4) Density-based Clustering: Existing works have used features or similarity matrices directly to form a community segment. Most of the studies use K-means clustering, non-negative matrix factorization-based clustering, nearest neighbor clustering, and others. These clustering methods are limited to the distance in Euclidean space, rather than considering the information-centric similarity. Furthermore, the computational complexity of the aforementioned methods increases with respect to the feature space, thus, hinder the scalability of the system [25]. In this regard, we adopt the density-based clustering approach [26] for community segmentation based on encoded embedding from similarity matrices. Furthermore, the density based clustering method has been extensively adopted to make the systems scalable. The only parameter that needs to be optimized is the number of communities; therefore, the number of communities with the highest partition density is considered accordingly.

C. Blockchain Layer

The blockchain network in this study is based on the characteristics of the Hyperledger Fabric framework, which consists of nodes such as committer, orderer, and endorser, accordingly. The transaction proposals submitted by vehicles are endorsed by the endorser node, the transformation of the transaction into blocks along with their packaging and sorting is performed by the orderer node, and the addition and validation of blocks to the database is carried out using the committer node. The transaction process mainly consists of the 4 steps, i.e. construction of transaction proposals by vehicles, emulation of transaction by endorser node, vehicles send the transactions to consensus service, and ordering of transactions via orderer nodes. These four steps are also used in the proposed CSB framework. Each of the entities in the CSB framework is defined below.

- vehicles (v): This entity is responsible for collecting data, $v_i \rightarrow i \in 1, ..., I$.
- UAVs (U): The entity is responsible for collecting vehicle data and sending the collected data to the AI Cloud, satellite storage, and the blockchain network, $U_j \rightarrow j \in 1, ..., J$.
- Authentication Certificate Provider (ACP): The entity issues a public key to each vehicle in the form of a digital certificate. The certificate can be used to prove the listing of the vehicle and the ownership of the public key later on.

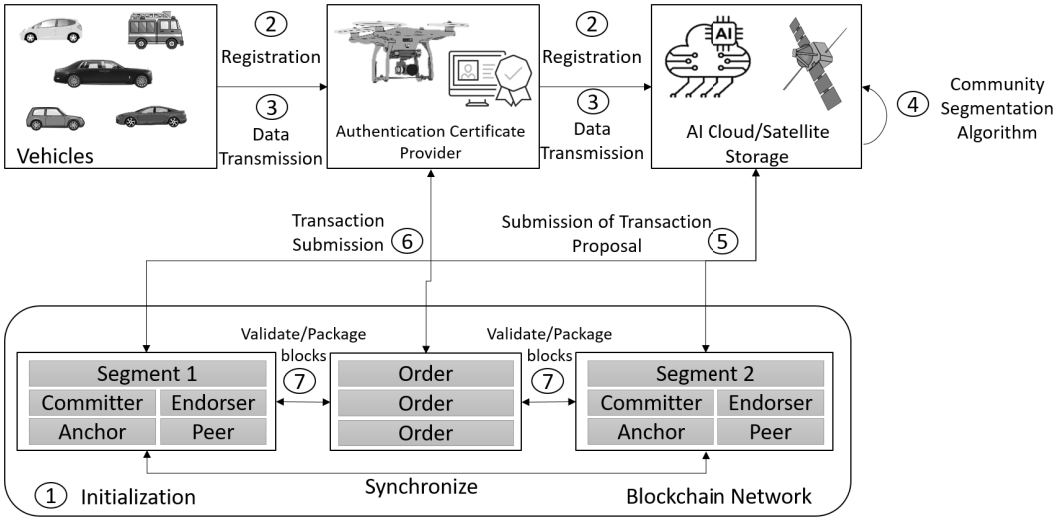

Fig. 3. Flow of Interaction in the proposed CSB framework

- AI Cloud/Satellite Storage: This is where the data from all the vehicles is sent. The collected data is then used for rendering the segmentation method in order to generate vehicle communities, and to transmit the outcome to the blockchain network.
- Endorser Node (EN): The transaction proposals succumbed by the vehicle are endorsed at this stage.
- Orderer Node (ON): The transactions are packaged and sorted into blocks at this stage.
- Committer Node (CN): Blocks are added and validated to the blockchain network at this stage.

1) Flow of Interaction: The flow of interaction concerning the blockchain, i.e. Hyperledger Fabric, has been adopted from the studies [27] and is shown in Figure 3. The flow of interaction comprises four stages, i.e., initialization, authorization, verification, and data sharing stage, respectively. The process of initialization mainly revolves around the generation of keys and IDs for UAVs and vehicles. The authentication service provider (ACP) selects two multiplicative groups Gr_1 and Gr_2 that are of the order of a prime number (PN). Subsequent to the selection mentioned above, a bilinear mapping $BM(\cdot)$ is performed such that $BM(Gr_1, Gr_1) = Gr_2$. The ACP then generates two $s \times r$ dimension matrices RM_u and RM_v along with two s dimensional column vectors CV_u and CV_v at random. These matrices and columns are represented in the form of linear equations, i.e. $RM_u \cdot \Phi_u = CV_u$ and $RM_v \cdot \Phi_v = CV_v$. The notation Φ_u and Φ_v represent the solutions to the aforementioned linear equations. The ACP calculates the columns Φ_{uj} and Φ_{vi} for each UAV U_j and vehicle v_i, respectively. The ACP then sends the aforementioned column vectors to the corresponding UAV and vehicle as proof of identity. The ACP subsequently generated two r dimensional column vectors Ω_u and Ω_v to calculate $ID_{Uj} = \Omega_u^T \Phi_{uj}$ and $ID_{vi} = \Omega_v^T \Phi_{vi}$. The randomly generated matrices, column vectors, and IDs, that is, $RM_u, RM_v, CV_u, CV_v, \Omega_u, \Omega_v, ID_{vi}$, and ID_{Uj}, are stored by the ACP. Information such as RM_v, CV_v, Ω_v and ID_{Uj} is sent to each UAV involved in the communication process, ID_{vi} is sent to each vehicle, whereas RM_u, CV_u, and Ω_u are sent to AI Cloud / Satellite Storage, accordingly. The ACP then generates two private keys PVK_{j1} and PVK_{j2} along with two public keys $PK_{j1} = PVK_{j1} \cdot Gr$ and $PK_{j2} = PVK_{j2} \cdot Gr$ in a random manner for each UAV. In order to initiate a communication with AI Cloud/Satellite Storage, ACP randomly selects two private keys and generates two public keys. Vehicles and UAVs then obtain these public parameters for a further communication process.

The purpose of the authorization stage is to authenticate the parties trying to communicate and exchange information with each other. UAV selects a random number (rnd) and computes the authentication parameters $AuP_1 = rnd.RM_v$ and $AuP_2 = rnd.CV_v$, accordingly. An authorization message is sent to the vehicle from UAV, that is, $msg1 = \{time_1, AuP_1, hash_{uj}\}$, where $time_1$ and $hash_{uj}$ are time stamps and one-way hash functions for the respective message. Vehicle then receives the authorization message from UAV and computes the parameter $AuP_{vi} = AuP_1.\Phi_{vi}$. As indicated in the study [27] that the parameters AuP_{vi}, AuP_2 should be equal, this implies that the hash functions will also be similar. Once the vehicle is legally verified by the UAV, the vehicle sends an authorization message, i.e. $msg2 = \{time_2, ID_{vi}, hash_{uj}\}$ to the UAV, however, this time the $hash_{uj}$ corresponds to $\{time_1||AuP_{vi}||ID_{uj}\}$ and $time_2$ is the time stamp of this message. Once the hash functions match, the identity parameters will be verified and the authorization will be completed. The verification stage ensures that the transmitted data are tamper-free. The verification process is quite similar to that of authorization, with only a slight modification in that the message is uploaded with its corresponding signature key. The rest of the process for computing parameters and comparing hash functions is the same. The study [27] could be referred for more details of the formulation and proof.

2) Data Sharing: The AI Cloud / Satellite Storage collects data from vehicles and performs the community segmentation algorithm to segment vehicles in different communities to

make the data sharing process efficient and scalable. The results from the segmentation of communities will be shared with the Hyperledger Fabric platform, accordingly. The process is summarized in the points below:

- AI Cloud/Satellite Storage collects the data from vehicles, performs authorization, and divides vehicles into different communities.
- A transaction proposal from AI Cloud/Satellite Storage for storing the segmentation results is initiated and submitted to the blockchain network.
- The results in the form of transactions are verified by the endorser nodes.
- If enough endorser nodes verify the transaction, the segmentation results are sent to the orderer nodes.
- The segmentation results are transformed into blocks by orderer nodes and broadcasted to the committer nodes.
- The committer nodes perform the final authorization and send the blocks generated at orderer nodes to the blockchain.
- If a UAV wants to access the segmentation results of the blockchain network, it should invoke a smart contract. A vehicle can also obtain its own community information from the UAV in a similar manner.

D. Evaluation Metrics

Most of the existing studies consider the contour coefficient method and a sum of squared error as the evaluation metric for the clustering method. Some studies have suggested that the former metric is unstable, while the latter one can be used reliably. Therefore, we consider the sum of squared error (SSE) metric to evaluate the density-based clustering approach. As we are dealing with the community segmentation which is based on the similarity matrices and encoded embedding, it is necessary to evaluate the reliability of the segmentation method. To do so, studies also employ sharing degree metrics to evaluate the quality of community segmentation. Sharing degree is jointly used with the error function such that the higher the rate of node similarity and lower error results in a high degree of sharing, accordingly.

IV. Experimental Results

A. Network Parameter

For the design of stacked autoencoders, we evaluated a variety of architectural parameters including the number of layers, the number of filters, the learning rate, the sparsity penalty term and the optimizer. On the basis of the number of experiments, we chose the parameters that achieved the best results in terms of SSE. The values for the parameters mentioned above are set to 5, 32-16-8, 0.01, 2.8, and SGD, respectively.

B. Simulation Setup

We generated 3 random datasets $RD_{1,\cdots,3}$ corresponding to the proportion of public data and the number of vehicles in a community. The details of the generated datasets are provided in Table 1, accordingly. The proportion of community data is represented by C_{data} and the number of vehicles in a community is represented by v_n. The data is represented in JSON format. The data features include vehicle name, sensor devices, communication protocol, application type, corresponding feature space, gateway type, and label.

TABLE I
Simulated Dataset Description

Dataset Name	v_n	C_{data}	data points	Ratio
RD_1	128	405	540	0.75
RD_2	128	225	500	0.45
RD_3	128	92	460	0.20

C. Evaluation of Community Segmentation

The main contribution of this study lies in the proposition of a community segmentation method for efficient data sharing within VSNs. In general, existing studies opt for the SSE metric to evaluate the clustering effect; however, current work also focuses on the efficiency of data sharing coupled with community segmentation. Therefore, we need to perform a joint evaluation of the sharing degree (SD) and SSE, accordingly. We first consider RD_1, to determine the number of communities in which the vehicles should be segmented. Vehicles are continuously added during the iterative process so that the resultant cluster may be updated accordingly. We present the rate of change in SSE and SD during the iterative clustering process in Figure 4 and Figure 5, respectively. The selection of an optimal number of communities is made based on the assumption that the optimal number of communities is the one before the SSE appears to be flat. As it is shown in Figure 4, the Elbow line connects the head and tail of the SSE curve (green) by a red line. We calculate the difference between the values of the Elbow line and SSE, the value with the maximum difference would be considered as the optimal number of communities as suggested by the Elbow method theory and existing studies [27]. The elbow method also provides the optimal number of communities concerning the SSE; however, we need to consider the number of communities that could help us optimize SD as well. Therefore, we consider the SD among three values, i.e. the optimal number of communities, one step forward and one step backward from the said number. We presume that the optimal number should not deviate from the region where the optimal number of communities lies. Amongst, these three values we consider the one with the highest sharing degree. As per the results presented, the number of communities with the highest difference between SSE and Elbow line is selected to be 7. Considering the aforementioned process, we choose 6, 7, and 8 communities to check the highest SD, which is selected to be 6, respectively. For the next set of experiments, we need to determine the number of iterations it would take to achieve the optimal Sharing degree. We performed this experiment for all three datasets, that is, RD_1, RD_2, and RD_3. The method is iterated so that the sharing degree gets improved while reducing the SSE, accordingly. The results for all three datasets are shown in Figure 6. The degree of sharing increases from 37 to 66, 10 to 26, and 3 to 11, when the ratio is selected

to 0.75, 0.45, and 0.20, respectively. The results indicate that the maximum Sharing degree for all three datasets is obtained around 300 iterations.

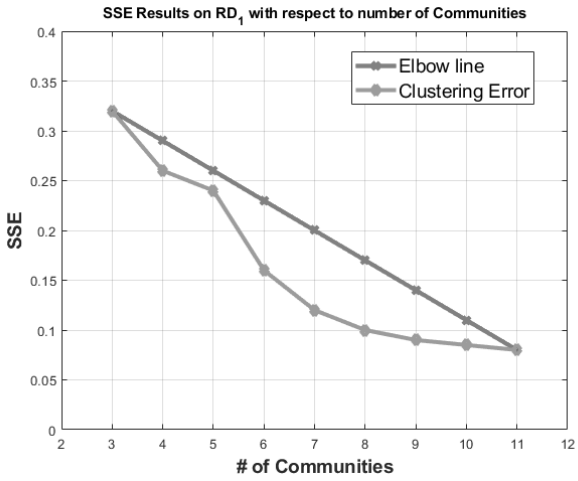

Fig. 4. The characteristics of Elbow line and SSE with respect to the number of communities

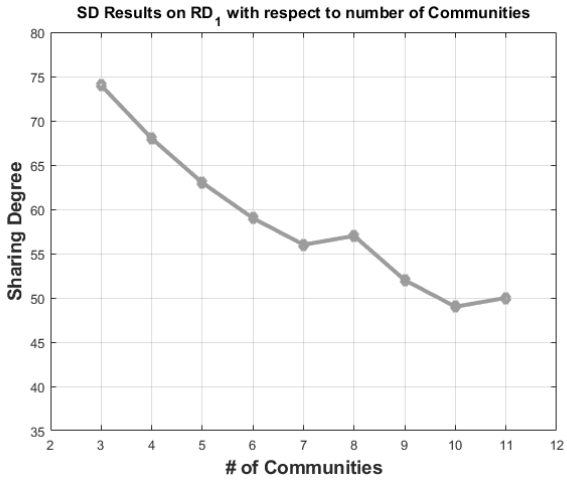

Fig. 5. The characteristics of Sharing degree with respect to the number of communities.

D. Evaluation of CSB framework for Data Sharing

The experiments in this study are carried out using an Intel Core i5 clocked at 3.4 GHz, 16GB of RAM, and a GTX 1080 GPU. The Docker was used to build the CSB framework for data sharing that includes 128 vehicle nodes, 1 AI/Satellite Cloud storage node, 1 Authentication Certificate Provider node, 1 order node, 1 community segmentation node, and 2 peer nodes. Vehicle data collected was stored directly at the community node. The implementation of a community node and a smart contract was carried out using Java and Python, respectively. The proposed CSB framework allows the vehicle to retrieve the data by querying the labels. Label queries are varied in order to measure the relationship between

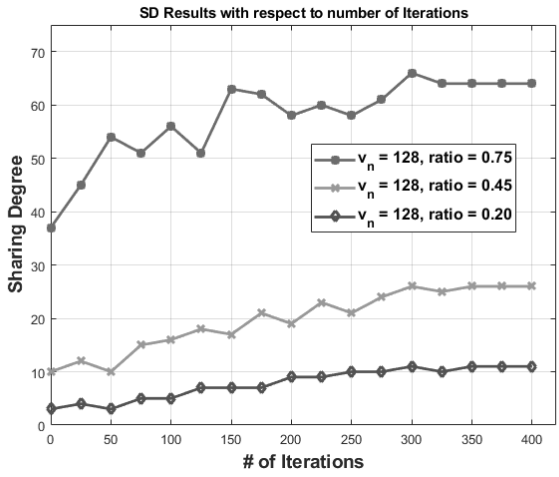

Fig. 6. Characteristics of SD with respect to the number of iterations

the time cost and query requests. At the time of querying, the similarity between the community node and the data from the queried label is computed to select the community segment for further data retrieval accordingly. We measure the time cost with respect to the number of query requests, as shown in Figure 7. It can be observed that the time cost increases with the number of query requests, which is in line with existing studies. However, increasing the number of query labels helps reduce computational complexity. We assume that the increased number of query labels narrows the search, resulting in a faster search response. Existing works also evaluate the performance of the data-sharing scheme through the time cost analysis and throughput, respectively. The former includes time for encryption and decryption, time for consensus among vehicles, i.e. verification, sorting, and endorsing time for chain code execution, and time for request initialization and response delivery. While the latter represent the total business volume, the number of blockchain system transactions per second also termed as transaction arrival speed, encryption algorithm, consensus algorithm, number of nodes, and server performance, accordingly.

We perform a simulation to process interaction with the blockchain system. The number of vehicles and the request arrival speed was set to [3, 7], and [20, 30], respectively. The results for the said simulation are shown in Figure 8 and Figure 9. It is revealed that increasing the number of vehicles helps to reduce the computation time. Similarly, the throughput increases with respect to the number of vehicles and arrival rate. This suggests that the proposed CSB framework can handle the query request within the acceptable time limit.

Existing studies also considered computational utilization in terms of CPU usage when increasing the number of communities. We report the experimental results for CPU utilization with the number of vehicles = 7 and the number of communities up to 1200 in Figure 10. It can be seen that even with 1300 communities, CPU utilization has not reached its maximum limit, indicating two aspects. The first is that the proposed CSB framework is resource- and energy-efficient, and the second is

that the proposed framework is capable of extending the scalability in terms of the blockchain network. It can be assumed from the obtained results that the proposed CSB framework can be used for a sustainable vehicular communication system, which is one of the main goals of European Commission for the future of the industries. Our results can be compared with an existing approach [4]. Although the parameters, application, and segmentation method are different for both of the studies, it can be analyzed that the proposed CSB can handle more queries in less time and yields better scalability. Furthermore, the proposed method also yields a better throughput. This study centers around beyond 5G network systems, the goal is to record improvement from existing works that could help in the development of real-world systems. The CSB framework is shown to improve system dynamics compared to existing systems with respect to the secure data sharing scheme.

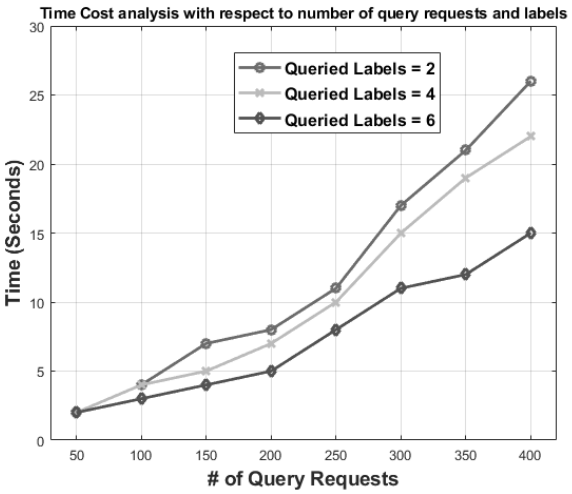

Fig. 7. The characteristics of computation time with respect to the number of query requests and query labels.

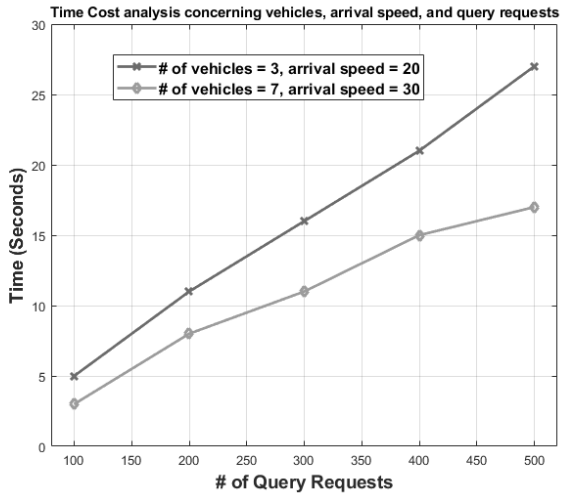

Fig. 8. Time Cost analysis for the proposed CSB framework

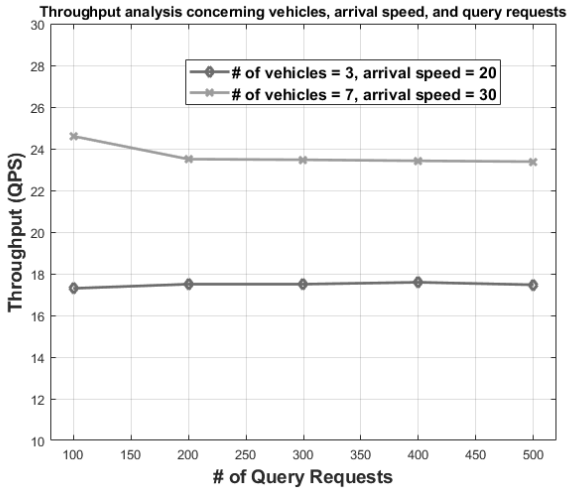

Fig. 9. Throughput analysis for the proposed CSB framework

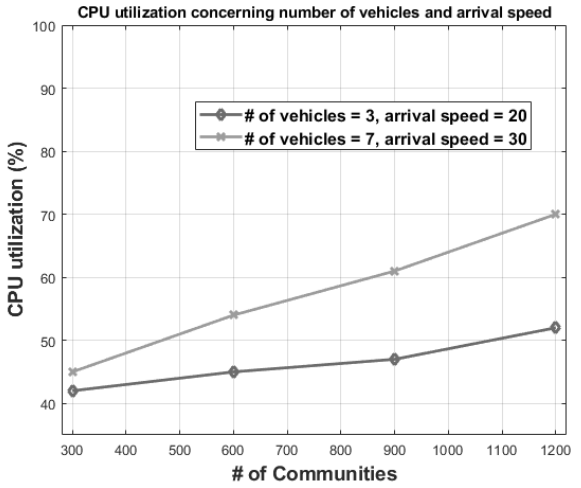

Fig. 10. The characteristics of CPU utilization with respect to the number of communities

V. Conclusion

This paper proposes a community segmentation and blockchain (CSB) based data sharing framework for vehicular social networks. We propose the use of similarity matrices considering structural similarity, modularity, and data compatibility followed by stacked autoencoders that transform similarity matrices into encoded embedding. The density-based clustering approach uses output embedding to segment the communities accordingly. We also implement a blockchain network that includes initialization, authentication, verification, and data sharing stages. We also propose a Hyperledger Fabric-based workflow for vehicular social networks to ensure secure data sharing, accordingly. We conducted extensive experiments to show the efficacy of the proposed CSB framework in terms of SSE, SD, time cost, computational complexity, throughput, and CPU utilization. The results show that the CSB framework not only achieves a higher degree of SD, lower computational complexity, and higher throughput, but

can also help increase the scalability of a blockchain network for vehicular networks.

In future work, we intend to increase the scope of the CSB framework for data and broadcasting security issues such as spoofing and replay attacks and to make the proposed framework lightweight to make it more realistic for current or future 5G networks.

REFERENCES

[1] J. Kang, R. Yu, X. Huang, and Y. Zhang, "Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2627–2637, 2018.

[2] J. He, K. Yang, and H.-H. Chen, "6g cellular networks and connected autonomous vehicles," *IEEE Network*, vol. 35, no. 4, pp. 255–261, 2021.

[3] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7729–7739, 2016.

[4] Y. Pu, T. Xiang, C. Hu, A. Alrawais, and H. Yan, "An efficient blockchain-based privacy preserving scheme for vehicular social networks," *Information Sciences*, vol. 540, pp. 308–324, 2020.

[5] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1366–1385, 2018.

[6] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2906–2920, 2019.

[7] C. Chen, C. Wang, T. Qiu, N. Lv, and Q. Pei, "A secure content sharing scheme based on blockchain in vehicular named data networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3278–3289, 2020.

[8] A. K. Das, B. Bera, S. Saha, N. Kumar, I. You, and H.-C. Chao, "Ai-envisioned blockchain-enabled signature-based key management scheme for industrial cyber-physical systems," *IEEE Internet of Things Journal*, vol. Early access, pp. 1–16, 2021.

[9] R. Gupta, A. Kumari, and S. Tanwar, "Fusion of blockchain and artificial intelligence for secure drone networking underlying 5G communications," *Transactions on Emerging Telecommunications Technologies*, p. e4176, 2021.

[10] L. Zhaojun, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2019.

[11] A. Abdallah and X. S. Shen, "Lightweight authentication and privacy-preserving scheme for v2g connections," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2615–2629, 2017.

[12] R. Hussain, D. Kim, J. Son, J. Lee, C. A. Kerrache, A. Benslimane, and H. Oh, "Secure and privacy-aware incentives-based witness service in social internet of vehicles clouds," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2441–2448, 2018.

[13] W. Zhao, Y. Qin, D. Gao, C. H. Fo, and H.-C. Chao, "An efficient cache strategy in information centric networking vehicle-to-vehicle scenario," *IEEE Access*, vol. 5, pp. 12 657–12 667, 2017.

[14] J. Ma, T. Li, J. Cui, Z. Ying, and J. Cheng, "Attribute-based Secure Announcement Sharing Among Vehicles using Blockchain," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 873–10 883, 2021.

[15] Y. Yao, X. Chang, J. Mišić, V. B. Mišić, and L. Li, "Bla: Blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3775–3784, 2019.

[16] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, "A secure charging scheme for electric vehicles with smart communities in energy blockchain," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4601–4613, 2019.

[17] Y. Jiang, X. Shen, and S. Zheng, "An Effective Data Sharing Scheme based on Blockchain in Vehicular Social Networks," *Electronics*, vol. 10, no. 2, p. p114, 2021.

[18] M. Sammarco, M. E. M. Campista, and M. D. de Amorim, "Scalable wireless traffic capture through community detection and trace similarity," *IEEE Transactions on Mobile Computing*, vol. 15, no. 7, pp. 1757–1769, 2016.

[19] F. Xia, L. Liu, B. Jedari, and S. K. Das, "Pis: A multi-dimensional routing protocol for socially-aware networking," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2825–2836, 2016.

[20] E. D. Raj, G. Manogaran, G. Srivastava, and Y. Wu, "Information granulation-based community detection for social networks," *IEEE Transactions on Computational Social Systems*, vol. 8, no. 1, pp. 122–133, 2021.

[21] L. Qiliang, S. Zhu, M. Deng, W. Liu, and Z. Wu, "A Spatial Scan Statistic to Detect Spatial Communities of Vehicle Movements on Urban Road Networks," *Geographical Analysis*, vol. 54, no. 1, pp. 124–148, 2022.

[22] K. Dev, S. A. Khowaja, P. K. Sharma, B. S. Chowdhry, S. Tanwar, and G. Fortino, "DDI: A Novel Architecture for Joint Active user Detection and IoT Device Identification in Grant-Free NOMA Systems for 6G and Beyond Networks," *IEEE Internet of Things Journal*, pp. 1–12, 2021.

[23] J. He, K. Yang, and H.-H. Chen, "6g cellular networks and connected autonomous vehicles," *IEEE Network*, vol. 35, no. 4, pp. 255–261, 2021.

[24] R. Xu, Y. Che, X. Wang, J. Hu, and Y. Xie, "Stacked autoencoder-based community detection method via an ensemble clustering framework," *Information Sciences*, vol. 526, pp. 151–165, 2020.

[25] S. A. Khowaja and P. Khuwaja, "Q-learning and LSTM based deep active learning strategy for malware defense in industrial IoT applications," *Multimedia Tools and Applications*, vol. 80, pp. 14 637–14 663, 2021.

[26] T. You, H.-M. Cheng, Y.-Z. Ning, B.-C. Shia, and Z.-Y. Zhang, "Community detection in complex networks using density-based clustering algorithm and manifold learning," *Physica A: Statistical Mechanics and its Applications*, vol. 464, pp. 221–230, 2016.

[27] J. Chi, Y. Li, J. Huang, J. Liu, Y. Jin, C. Chen, and T. Qiu, "A secure and efficient data sharing scheme based on blockchain in industrial internet of things," *Journal of Network and Computer Applications*, vol. 167, pp. 1–10, 2020.