# A Trust Model for Edge-Driven Vehicular Ad Hoc Networks Using Fuzzy Logic

Md. Mahmudul Hasan, Mosarrat Jahan, *Member, IEEE*, and Shaily Kabir, *Member, IEEE*

***Abstract*— Trust establishment among vehicles is essential for vehicular ad hoc networks (VANETs) as it directly impacts the security and privacy of vehicular communication. Many trust estimation approaches have been introduced, however, they often suffer from ensuring effective trust for vehicles. In fact, existing approaches do not involve all malevolent properties of vehicles in trust computation and can not properly handle the content tampering attack, which eventually affect the accuracy of the estimated trust. Moreover, most of them do not consider the uncertainty of VANET arising from vehicles' mobility, their inaccurate/incomplete data dissemination, and the wireless communication channels, which also affects the reliability of the trust estimation. To address these limitations, this paper proposes a fuzzy logic-based approach to estimate vehicles' trust. The new approach considers three trust factors, captured by fuzzy sets, to model malicious properties of a vehicle. Further, it involves a new data-centric parameter to capture the impact of content tampering on trust evaluation. In addition, the new approach includes an inter-edge trust transfer mechanism to carry forward a vehicle's trust when it switches to a new edge server to ensure a seamless operation in VANETs. We evaluate the performance of the proposed scheme against the state-of-the-art approaches using both synthetic and real-world datasets. The experimental results reveal that it outperforms existing schemes in detecting malicious vehicles with higher recall, precision, and accuracy. Further, the new scheme reduces end-to-end delay and messages per data packet compared to other schemes.**

***Index Terms*— Vehicular ad hoc network (VANET), trust, fuzzy logic, V2I communication, edge computing.**

## I. Introduction

VEHICULAR Ad hoc Network (VANET) plays a significant role in intelligent transportation system (ITS) which is essential for building a smart city [1]. VANET provides information flow for efficient traffic management, road safety, and better travel experience by establishing link among three components—static infrastructure-based roadside units (RSUs), edge servers, and moving vehicles [2], [3]. Being a dynamic topology and time-critical network, an efficient VANET operation largely depends on the correctness of disseminated information and swift response to vehicles' queries. However, the presence of malicious vehicles in VANETs and the lossy nature of wireless network fail to ensure consistent receiving of accurate information and often increase query response time [4], [5], causing severe consequences such as security and privacy hamper and an increase of road accident [4], [5].

In VANET communication, a vehicle behaves maliciously accidentally through component malfunction and node compromise by adversaries or intentionally which often disrupts normal functionalities of the network [6]. In both cases, it disseminates false information, modifies packet content, and drops/delays data packet transmission [6] which result in various security and privacy attacks for the network, e.g., packet drop/delay-based black hole attacks, grey hole attacks, content alteration-based message tampering, man-in-the-middle attacks, and packet injection-based identity impersonation attacks [7], [8]. Hence, it is essential to identify and separate malicious vehicles from the good ones in VANETs and minimize their adverse effect.

Trust estimation of vehicles is a cost-effective and efficient method to identify malicious ones which systematically analyzes vehicles' dynamic behavior through different trust factors such as packet forwarding ratio, recommendation from neighbor vehicles, and vehicular interaction [9], [10], [11], [12]. Nonetheless, accurate trust estimation is often challenging because of the unpredictable behavior of vehicles as well as wireless networks. Present trust estimation approaches inspect only the packet forwarding ratio—presenting vehicles' behavior to drop/delay packets—while ignoring two other important features of malicious vehicles: *content alteration* and *insertion of false information*. Further, they are often indecisive about the actual content of a data packet when receiving its multiple copies with some copies tampered by malicious relay vehicles. As a result, they can compute and assign low trust value for a good vehicle just for passing an altered packet while high trust value for the actual malevolent one [10], [11].

Trust computation is further affected by the uncertain and unreliable VANET communication [10]. In fact, variation in speed and density of vehicles in roads along with unpredictable environment and road condition introduce such uncertainty in VANET which often causes loss of disseminated information [10], [13]. In addition, vehicles' transient contact time, faulty components and sensors contribute to incompatible factor values and imprecise/incomplete data on network which again create ambiguity and inaccuracy in trust computation [10]. Existing researches apply fuzzy logic on trust factors to deal with such uncertain and lossy behavior

Manuscript received 26 June 2022; revised 26 March 2023 and 26 June 2023; accepted 4 August 2023. Date of publication 11 September 2023; date of current version 29 November 2023. The Associate Editor for this article was H. Jiang. *(Corresponding author: Mosarrat Jahan.)*

The authors are with the Department of Computer Science and Engineering, University of Dhaka, Dhaka 1000, Bangladesh (e-mail: 2014-616-631@student.cse.du.ac.bd; mosarratjahan@cse.du.ac.bd; shailykabir@cse.du.ac.bd).

Digital Object Identifier 10.1109/TITS.2023.3305342

of vehicular networks [9], [10], [11], [14]. Xia et al. [11] focused on vehicles' packet drop/delay property and used packet forwarding ratio, recommendation, and interaction with neighbor vehicles to measure fuzzy trust values. Further, Guleng et al. [10] proposed a fuzzy trust estimation scheme that uses packet drop/delay feature along with content alteration of malicious vehicles. However, these two approaches face accuracy problem in estimating packet forwarding ratio as they do not consider the impact of vehicles' velocity. Moreover, they fail to differentiate between malevolently modified and authentic message and erroneously reduce trust values of vehicles.

To overcome the aforementioned shortcomings, we initially put forward a novel fuzzy logic-based trust estimation scheme [15] for VANETs considering all malicious behavior of vehicles and the uncertainty of wireless environment. The new scheme used packet drop, content alteration and false packet injection properties of vehicles as trust factors while computing the trust. Further, it was designed to automatically readjust the trust of a vehicle when its trust score was falsely reduced due to the content alteration attack. Its performance was evaluated using synthetic dataset where it outperformed other approaches by producing vehicles' trust with high accuracy. However, our proposed scheme could not propagate the trust value of a vehicle from one edge server to another but recomputed its trust value from the scratch. During the recomputation time, the vehicle's trust score was undefined to the interacting vehicles, thus creating problem for them to decide whether it is a reliable vehicle or not in this region.

To minimize this limitation, in this paper, we propose a refinement of our scheme by incorporating an inter-edge trust transfer mechanism which ensures a coherent operation within VANETs. Specifically, the proposed refined scheme transfers the trust score of a vehicle from one edge server to another by employing the wired and wireless connectivity among edge servers, as a result, no indecisive situation can arise and the VANET operation can proceed smoothly. We evaluate the new model using both synthetic and real-world datasets e.g., *Vehicular Reference Misbehavior Dataset (VeReMi)* [16] and its extension, *extended VeReMi* [17]. In addition. we extend the performance analysis of this refined model on the synthetic dataset with an analysis of additional network performance metrics. We also demonstrate the impact of the new scheme on different VANET routing protocols. We note that in all experiments, we compare the proposed refined model against two well-known trust estimation schemes—Guleng et al.'s scheme [10] and Xia et al.'s scheme [11]. Major contributions of this paper are as follows:

- A fuzzy logic-based trust estimation mechanism is proposed to deal with the unreliability of trust estimation process arose from malicious behavior of vehicles and the uncertain and unpredictable nature of VANET.
- Uncertainty in trust estimation is handled by using three fuzzy trust factors—*Packet Drop Factor (PDF)* presenting the malicious property of dropping/delaying data packets, *False Packet Injection Factor (FPIF)* expressing the evil property of generating false information, and *Content Alteration Factor (CAF)* indicating the tendency of tampering the original message content.
- A new data-centric factor—*Network Topology Factor (NTF)* is introduced to address the impact of message tampering attacks in trust calculation.
- An inter-edge trust transfer mechanism is proposed to carry forward the trust value of a vehicle when it switches to a new edge server's domain.
- Performance analysis with a synthetic dataset on various network settings and routing protocols reflects the superiority of the proposed model, which is obtained by assessing the uncertain behavior of vehicles more accurately, detecting malicious vehicles with higher precision and recall, and enduring increasing network complexity. Our scheme reduces messages per data packet by nearly 37% and end-to-end delay by approximately 40% compared to the state-of-the-art.
- Performance analysis using the VeReMi dataset and its extension reveals that our scheme can detect around 36% and 31% more malicious vehicles than the existing research works in DoS and data replay attacks.

The rest of the paper is organized as follows. Section II discusses the related works on trust evaluation and Section III presents the background information. Section IV introduces the proposed system model, while Section V describes the detailed operation of the proposed scheme. Section VI presents the results of the experimental evaluations. Finally, we conclude the paper in Section VII.

## II. Related Work

Various security measures such as public key infrastructure (PKI) [18], digital signature [19], and trust estimations [20] have been proposed to deal with security attacks in wireless communication networks. The PKI and digital signature-based authentication mechanisms can successfully remove external security attacks but fail to handle internal attacks in VANETs caused by authenticated vehicles behaving maliciously for component malfunctions and node compromise [6], [21]. Further, they do not consider the dynamic and lossy environment of VANETs and are computation-intensive, thus not suitable for dynamic-nature VANETs [11], [21].

In contrast, trust estimation based on vehicle cooperation and behavioral traits is an efficient and faster way to assess vehicles' trustworthiness, which can mitigate internal attacks in VANETs [11], [12], [20]. Different entity- and data-centric factors are used to compute trust, where the former factor assesses the behavior of vehicles using vehicle and engine condition, packet transmission rate, responsibility of reporting observed events, interaction with neighbors, recommendations from prior experience, and many more [3], [9], [10], [11], [12], [13], [22], [23] while the latter one validates the integrity of received information with correctness ratio of transmitted data packets and plausibility level which verifies location and time of source vehicles [6], [9], [10], [21], [24], [25]. Despite validating the

received packets, the data-centric factors still face limitations in capturing the effect of malicious relay vehicles on message content. Moreover, they do not explicitly examine vehicles' false packet injection properties. In the proposed scheme, we integrate both entity- and data-centric factors along with the newly introduced three trust factors to effectively handle all malicious properties of vehicles and to accurately compute the trust values for vehicles.

As VANET operations are uncertain and unreliable, many researches already involved fuzzy logic on trust factors for effective trust computation. Guleng et al. [10] proposed a fuzzy trust estimation scheme for decentralized VANET, incorporating packet transmission ratio, responsibility, and correctness ratio for direct trust (trust computation of one-hop neighbors) calculation and a Q-learning approach to calculate the indirect trust (trust computation of vehicles more than one hop away). Mahmood et al. [24] introduced a hybrid trust management scheme integrating trust value with resource availability of vehicles. Xia et al. [11] proposed a fuzzy trust-based multicast routing protocol combining packet transmission ratio, activity factor, and recommendation credibility. Later, they incorporated historical information and neighbors' feedback with recommendations of vehicles in another research work [12]. Zhang et al. [25] also used historical data with social factors to design a fuzzy trust management scheme. Besides, Soleymani et al. [14] presented a fuzzy trust estimation scheme using plausibility, experience, and vehicle type that utilizes the huge computation power of edge servers to perform trust evaluation. Later, they proposed a fuzzy logic-based trust model that examines the plausibility and accuracy of disseminated information, and previous experience [9]. Apart from that, An et al. [3] proposed a fuzzy trust-based packet transmission mechanism utilizing vehicle velocity, distribution, and channel conditions. Noted that all of these fuzzy models compute trust values for vehicles without exploring all hostile properties of vehicles, which implies their limitation on reliably distinguishing between trusted and malicious vehicles, arising a big question on the accuracy of their trust computations.

In addition, existing trust estimation schemes cannot properly handle content tampering attack (i.e., inability to realize correct content of data packets) which directly hampers the correct functioning of VANETs. To deal with this, Chen et al. [6] proposed a heuristic and optimal decision algorithm considering network topology to distinguish authentic message content from the tampered one. On the other hand, Al Zamil et al. [26] applied the Hidden Markov Model (HMM) in binary classification to detect false alarms while Al-Otaibi et al. [5] utilized location information to verify the validity of received information. In addition, Radak et al. [27] applied distributed data fusion and self-stabilizing algorithm to deal with unreliable data sources and Raya et al. [28] used evidence evaluation techniques to ensure the validity of received information.

In summary, existing trust estimation works combine various entity- and data-centric factors to identify malicious vehicles but are unable to correctly analyze all significant malicious properties. They do not examine the impact of malicious relay vehicles on message content and thus cannot guarantee data integrity and accurate trust values for vehicles. In contrast, the proposed model (discussed in Section V) evaluates all significant malicious properties of vehicles and introduces three fuzzy trust factors to capture them. To guarantee accurate trust estimation, it also proposes a new data-centric factor to ensure the integrity of received information and to adjust falsely reduced trust values of vehicles to diminish the impact of malicious relay vehicles.

TABLE I
LIST OF NOTATIONS

| | | | |
|---|---|---|---|
| *FF* | Forwarding Factor [10, 11] | *NTF* | Network Topology Factor |
| *MF* | Monitoring Factor [10] | *PDF* | Packet Drop Factor |
| *SF* | Swiftness Factor [3] | *FPIF* | False Packet Injection Factor |
| *TF* | Trustee Factor [24] | *CAF* | Content Alteration Factor |
| *AF* | Activity Factor [11] | *RC* | Recommendation Credibility [11] |

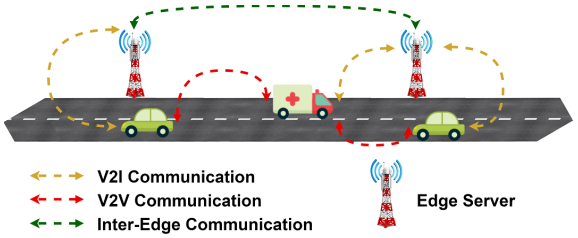


Fig. 1. System model of the proposed scheme.

## III. PRELIMINARIES

*1) Forwarding Factor (FF)*: It is an entity-centric factor showing a vehicle's behavior in transmitting packets [10], [11]. A vehicle $i$ computes the $FF$ of a one-hop neighbor vehicle $j$ as the ratio of the number of packets received from $j$ and the average number of packets received from its one-hop neighbor vehicles.

*2) Monitoring Factor (MF)*: It is also an entity-centric factor describing a vehicle's behavior to report observed events [10]. A vehicle $i$ determines the $MF$ of a one-hop neighbor vehicle $j$ as the quotient of the number of reported events by $j$ and the average number of events detected by $i$ and informed by its one-hop neighbor vehicles.

*3) Swiftness Factor (SF)*: It is an entity-centric factor too denoting the relative velocity of a vehicle [3]. A vehicle $i$ computes the $SF$ of a one-hop vehicle $j$ as the ratio of the speed of $j$ and the highest speed of its one-hop neighbors.

*4) Trustee Factor (TF)*: It is a data-centric factor indicating the accuracy in transmitting authentic information [10]. A vehicle $i$ computes the $TF$ of a one-hop neighbor vehicle $j$ as the ratio of correct packets and the total number of packets received from $j$. The information obtained from the maximum number of received packets denotes the correct status of an event [10]. Table I presents a list of notation used in this paper.

## IV. PROPOSED SYSTEM MODEL

Figure 1 presents our proposed model, which incorporates vehicles and edge servers. Vehicles usually observe and gather

information on road conditions, environment status, and undesirable events while traveling and disseminate data packets to neighbor vehicles and edge servers through Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications, respectively. In the proposed system model, vehicles calculate the trust factor values of all neighboring one-hop vehicles by observing their behavior. Besides, they review data consistency and integrity of exchanged information and transmit required data to respective edge servers for accurate trust estimation of vehicles. We assume each vehicle in the VANET contains a unique, authentic identification number [6]. Although authenticated vehicles are expected to work honestly, they may show malicious properties over time in a number of situations, including selfish behavior shown by vehicles, components and sensors malfunction, attacks by adversaries, etc. [29]. Malicious vehicles show three main hostile properties – insert false data packets, drop or delay packets, and modify the original content [6].

Edge servers are semi-trusted localized cloud servers providing computation, communication, and storage support to vehicles [4]. In the proposed model, edge servers accumulate diverse trust factors from vehicles, compute new trust values and update previously assigned trust values of vehicles. We assume that vehicles communicate with edge servers using SSL channels. Each edge server communicates with neighboring edge servers using inter-edge communication and exchange information, including relevant trust values of vehicles moving from one edge server's jurisdiction to another [1]. Edge servers are connected through secure wired channels, or wireless connections in a standard backbone network [25].

Data packets transmitted by vehicles include the source vehicle's ID and position, packet type, perceived information, and an integrity-protected list of relay vehicles [10]. In VANETs, every vehicle in transmission paths includes its ID to the integrity-protected path list [6]. We assume that exchanged packets have a vector of events, where a binary value represents each event, either 0 or 1, indicating the occurrence of a single event [6]. Besides, the source vehicle restricts the hop counts to 5 while broadcasting data packets due to the lossy wireless environment, and limited significance of VANET information [6], [10], [12], [13]. We also assume that the VANET system is time-synchronized [30].

## V. Proposed Scheme

### A. Design Rationale

The proposed model uses fuzzy sets to represent the newly introduced trust factors depicting the malicious properties of a vehicle. Among them, PDF reflects the behavior of dropping/delaying data packets by measuring the accurate packet transmission rate. In the lossy vehicular network, high-speed vehicles drop more data packets than vehicles with a lower velocity due to the lack of dedicated communication channels. Existing *FF* [10], [11] disregards the impact of velocity and thus, reduces the trust values of high-speed vehicles. We apply both *FF* and *SF* in *PDF* computation

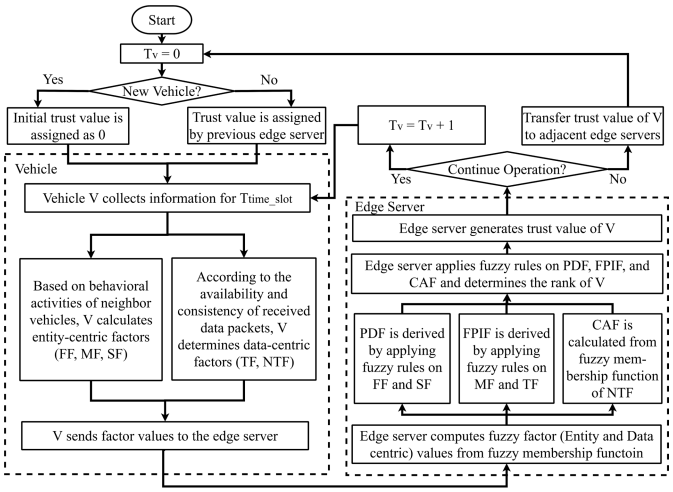


Fig. 2. Flow diagram of the proposed scheme.

to overcome this shortcoming. Besides, *FPIF* inspects the behavior of generating incorrect and invalid data packets. Though vehicles with high monitoring ratio detect and transmit more data packets and are anticipated to be trustworthy, they may transmit false information intentionally to gain unfair advantages or due to component malfunctions. We handle this issue by introducing the *FPIF* that justifies the validity of events claimed in *MF* against the transmission ratio of valid packets *TF* of a vehicle. Moreover, *CAF* analyzes the behavior of a malicious vehicle in altering information of data packets. Our scheme measures *CAF* through *NTF* that detects hostile relay vehicles in transmission paths and adjusts the falsely reduced trust values of honest vehicles for transferring packets tampered by the evil relay vehicles. Finally, we employ a fuzzy logic-based trust estimation algorithm utilizing the proposed fuzzy trust factors to cope with the incomplete, imprecise data and the uncertain VANET environment.

### B. Overview of the Proposed Scheme

The proposed trust model splits the time into different time slots, $T$. At the beginning of each $T$, vehicles observe and gather information utilizing various sensors for a pre-defined time duration, $T_{time\_slot}$. After that, vehicles evaluate different entity-centric factors – *FF*, *MF*, *SF*, and data-centric factors – *TF*, *NTF* for their one-hop neighbors and send them to respective edge servers for further analysis. After receiving varying factor values from vehicles, an edge server considers the mean values to calculate – *PDF*, *FPIF*, and *CAF* for a particular vehicle and then determines the accurate trust values for vehicles by applying a fuzzy logic-based trust evaluation algorithm. The flow diagram shown in Fig. 2 reflects the working principle of both a vehicle and edge server and the trust estimation mechanism of the proposed scheme. At initial, the trust value for a newly introduced vehicle in VANET is set to 0 [3], [23] while others maintain previously assigned trust values. A vehicle sets the time counter $T_v$=0 whenever it enters into the jurisdiction of a new edge server. After

the end of $T_v$=0, the edge server calculates trust values for vehicles from the values of the received factors and assigns trust values to new vehicles and updates for existing ones. Once the trust value of a vehicle is determined, in the later time phases, factor values are only used to revise the existing trust value. A vehicle updates $T_v$ if it continues its operation in the same region. When a vehicle switches to a new edge server, related information, including the trust value of the vehicle, is transferred to the neighboring edge servers by applying inter-edge communication.

### C. Trust Factors Calculation

*1) Entity-Centric Trust Factors:* A vehicle collects information for $T_{time\_slot}$ and evaluates three entity-centric trust factors – *FF* [10], [11], *MF* [10], and *SF* [3], [23] for each neighboring vehicle, reflecting the behavior of packet transmission ratio, event detection and reporting tendency, and velocity of vehicles, respectively.

*2) Data-Centric Trust Factors:* The proposed scheme computes data-centric trust factors of one-hop neighbor vehicles through the existing *TF* [10] and newly introduced *NTF* based on the location of source vehicles. If source vehicles remain within the transmission range of the trustor vehicle, we apply *TF* to regulate the integrity of the received data packets. When source vehicles stay more than one hop distance, *NTF* is applied to calculate the data-centric factor.

- Trustee Factor (*TF*), as discussed in Section III, exhibits the proportion of accurate data transmission by a vehicle [10]. The proposed scheme applies *TF* to the one-hop neighbor vehicles reporting observed events. The credibility of the informed event is ensured by examining the trustor vehicle's sensed information and the information received from other one-hop neighbor vehicles [10].
- Network Topology Factor (*NTF*) determines the data-centric trust factors for one-hop neighbors that act as relay vehicles for forwarding data packets transmitted from distant source vehicles. *NTF* detects malicious relay vehicles in transmission paths and diminishes their negative impacts on trust estimation of vehicles.

Vehicles inform the occurrence of various events by broadcasting data packets to neighbor vehicles. These packets transmit from source vehicle to destination through multiple transmission paths where the content of the packets might be modified by malicious relay vehicles present in the paths. Hence, a vehicle might receive numerous copies of a data packet containing erroneous information, creating uncertainty in deciding the correct content from the received messages. Though *TF* validates the content of messages generated from one-hop neighbor vehicles, it cannot accurately verify the content originated from distant source vehicles as *TF* does not consider the underlying network topology and the impact of malicious relay vehicles on the message content. The proposed scheme introduces *NTF* to overcome this shortcoming.

*3) NTF Calculation:* The proposed scheme applies the heuristic decision algorithm [6] to detect the correct content of messages where packets are originated beyond the one-hop range. Based on the output decision variable $D$, we compute the set of nonaffected vehicles $NV$, the set of affected vehicle $AV$, and the set of malicious vehicles $MV$ using Algorithm 1 proposed in our earlier work [15]. *NTF* takes measures to lessen the impact of content alteration attacks by the vehicles in $MV$ in the trust computation of vehicles in $AV$. A vehicle $i$ calculates the *NTF* value of one-hop vehicle $j$ as follows:

$$NTF_{ij} = \frac{P_{true_j} + P_{affected_j}}{P_{total_j}} \tag{1}$$

where, $P_{total_j}$ is the total number of paths containing vehicle $j$, $P_{true_j}$ is the number of paths where $j$ delivers accurate data and $P_{affected_j}$ is the number of affected paths where $j$ delivers incorrect information due to content alteration by malicious relay vehicles in $MV$. Vehicle $i$ computes $P_{total_j}$, $P_{true_j}$, and $P_{affected_j}$ from the network topology information obtained from the integrity-protected path lists of received packets. Vehicle $i$ estimates different *NTF* values for a neighbor $j$ for multiple events and takes the mean value of all to determine the final *NTF* value of $j$. If $j$ is a malicious vehicle, $NTF_{ij} = 0$. The detailed description of computing *NTF* can be found in [15]. For the page limitations, we omit the thorough explanation.

### D. Fuzzy Trust Factors Computation

*1) Membership Values Calculation:* Each neighboring vehicle sends different sets of (*FF*, *MF*, *SF*, *TF*, and *NTF*) values to the edge server for a specific vehicle. Hence, an edge server averages all the values for a particular factor to obtain a final value and applies this value to the corresponding fuzzy membership function to obtain membership values. Figures. 3(a) ∼ (d) represent fuzzy membership functions used in the proposed scheme for *FF*, *TF*, *MF*, and *SF*, respectively. These membership functions are determined by the expert insight following [3], [10].

*2) Packet Drop Factor (PDF):* The proposed scheme derives the $PDF$ of a vehicle from the fuzzy values of $FF$ and $SF$ according to the fuzzy IF/THEN rules given in Table II. These rules consider the impact of velocity on the packet transmission ratio. A vehicle that exhibits $FF = \{Strong\}$ in any situation gains $PDF = \{Strong\}$ for transmitting sufficient data packets. A vehicle with weak velocity ($SF$) acquires a larger interaction time and is thus anticipated to have a strong packet forwarding ratio ($FF$). Hence, a vehicle with $SF = \{Weak\}$ displaying other values for $FF$ except $\{Strong\}$ is considered as malicious, leading to $PDF = \{Weak\}$. Likewise, a vehicle with $SF = \{Moderate\}$ and $FF = \{Weak\}$ is unconvincing. Vehicles that achieve strong $PDF$ values are reliable in transmitting packets, while weak values mark the selfish behavior of vehicles that may drop packets. When multiple rules are applicable for a vehicle, the proposed scheme uses min-max method [13] to determine the trust factor.

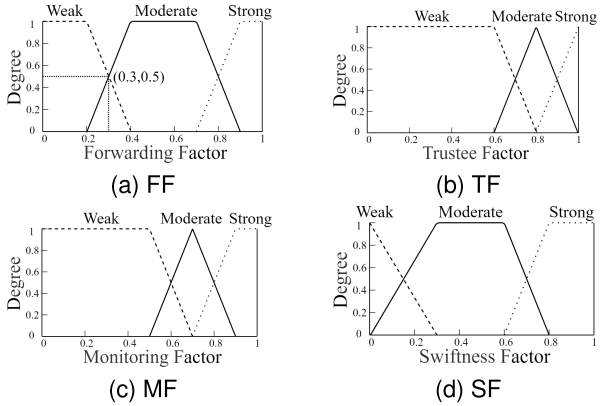


Fig. 3. Fuzzy membership function for FF, TF, MF, and SF.

*3) False Packet Injection Factor (FPIF):* It uses the fuzzy values of $TF$ and $MF$ to classify vehicles as *standard* or *hostile* following to the rules presented in Table III. A vehicle that reports more events with $MF = Strong$ but shows low correct packet forwarding ratio, $TF = Weak$, indicates that it maliciously inserts false packets. Therefore, in Table III, $MF > TF$ leads to $FPIF = \{Hostile\}$.

*4) Content Alteration Factor (CAF):* It uses the mean $NTF$ values of vehicles in the fuzzy membership function presented in Fig. 4(a) to classify vehicles as *standard* or *hostile*.

### E. Trust Value Calculation and Update

*1) Rank Calculation of Vehicles:* An edge server, after calculating the fuzzy values of *PDF*, *FPIF*, and *CAF*, determines the rank of vehicles by following the fuzzy IF/THEN rules shown in Table IV, where the output ranks of vehicles are classified as *below*, *low*, *average*, *medium*, *high*, and *perfect*. Table IV shows the fuzzy rules to determine the rank of vehicles based on the priority-assigned order of fuzzy factor values reflecting the malicious properties of a vehicle. Secure VANET operations require accurate transmission of data packets and thus, highest priority is assigned to $PDF$. *FPIF* verifies the credibility of a vehicle as a source of information, and *CAF* reveals the honesty as a relay vehicle. Hence, higher priority is assigned to *FPIF* than the *CAF* after the *PDF*. A vehicle with *PDF*={Strong}, *FPIF*={Standard}, and *CAF*={Standard} maintains adequate packet transmissions and secures the flow of trusted information and thus, is marked as *Perfect* vehicle according to Table IV. Similarly, other rules consider the remaining values of trust factors to generate ranks according to the priority-assigned order.

*2) Trust Value Calculation:* After calculating the fuzzy output membership values, an edge server applies the Center of Gravity (COG) method on the overall area of the membership function distribution to precisely determine the trust values of vehicles [3], [13]. Due to malicious vehicles and defective sensors, fuzzy membership values from different sources may show enormous differences. To handle this disparity and obtain

TABLE II
RULES FOR PDF VALUES

| IF/THEN Rules | FF | SF | PDF |
|---|---|---|---|
| Rule 1 | Strong | Strong | Strong |
| Rule 2 | Strong | Moderate | Strong |
| Rule 3 | Strong | Weak | Strong |
| Rule 4 | Moderate | Strong | Strong |
| Rule 5 | Moderate | Moderate | Moderate |
| Rule 6 | Moderate | Weak | Weak |
| Rule 7 | Weak | Strong | Moderate |
| Rule 8 | Weak | Moderate | Weak |
| Rule 9 | Weak | Weak | Weak |

TABLE III
RULES FOR FPIF VALUES

| IF/THEN Rules | TF | MF | FPIF |
|---|---|---|---|
| Rule 1 | Strong | Strong | Standard |
| Rule 2 | Strong | Moderate | Standard |
| Rule 3 | Strong | Weak | Standard |
| Rule 4 | Moderate | Strong | Hostile |
| Rule 5 | Moderate | Moderate | Standard |
| Rule 6 | Moderate | Weak | Standard |
| Rule 7 | Weak | Strong | Hostile |
| Rule 8 | Weak | Moderate | Hostile |
| Rule 9 | Weak | Weak | Hostile |

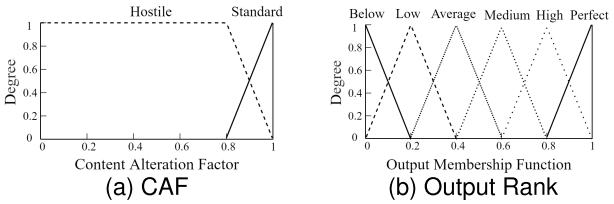


Fig. 4. Fuzzy membership function for CAF and output rank.

acceptable trust values, we choose CoG as it always maintains a stable equilibrium. This method returns a crisp value, the center of the area under the curve. The overall area is divided into several sub-areas according to the membership function distribution to establish a combined control. Each sub-region's area and center of gravity are determined and then applied in the defuzzification process to compute the crisp value. Figure 4(b) shows the output membership function [3], [10]. If median values for output results ($d_1$, $d_2$,..., $d_n$) and corresponding degree values ($w_1$, $w_2$,..., $w_n$) then $x$ coordinate of the centroid in the output membership function represents trust value of a vehicle according to (2).

$$COG(x) = \frac{d_1 w_1 + d_2 w_2 + \ldots + d_n w_n}{w_1 + w_2 + \ldots + w_n} \tag{2}$$

*3) Trust Value Adjustment:* An edge server calculates the trust values of new vehicles at the end of $T_v = 0$, while it readjusts the trust values of other vehicles in the later phases using (3).

$$T_{new}(j) = \alpha \times T_{fuzzy}(j) + (1 - \alpha) \times T_{old}(j) \tag{3}$$

where $T_{new}(j)$ is the new trust value of vehicle $j$, $T_{fuzzy}(j)$ is the fuzzy logic-based trust value of vehicle $j$, $T_{old}(j)$ is the earlier trust value, and $\alpha$ is a smoothing factor reflecting

TABLE IV
FUZZY RULE BASE: RANK OF VEHICLES

| IF/THEN Rules | PDF | FPIF | CAF | Rank |
|---|---|---|---|---|
| Rule 1 | Strong | Standard | Standard | Perfect |
| Rule 2 | Strong | Standard | Hostile | Medium |
| Rule 3 | Strong | Hostile | Standard | Average |
| Rule 4 | Strong | Hostile | Hostile | Low |
| Rule 5 | Moderate | Standard | Standard | High |
| Rule 6 | Moderate | Standard | Hostile | Medium |
| Rule 7 | Moderate | Hostile | Standard | Average |
| Rule 8 | Moderate | Hostile | Hostile | Low |
| Rule 9 | Weak | Standard | Standard | Average |
| Rule 10 | Weak | Standard | Hostile | Low |
| Rule 11 | Weak | Hostile | Standard | Below |
| Rule 12 | Weak | Hostile | Hostile | Below |

a co-relation between new and old values while emphasizing the new observed values.

### F. Inter-Edge Trust Transfer

As vehicles travel across multiple edge servers, a standard backbone of a wired or wireless network internally connects edge servers to transfer information about the moving vehicles, workloads, and unsatisfied requests [1], [25]. Based on periodic beacon messages of vehicles, an edge server determines the current position, moving direction, and dwelling period of vehicles [1]. Besides, route information delivered by a vehicle utilizing a digital map, GPS navigator, neighbor vehicles' information, and current traffic condition [31] helps the corresponding edge server to identify exactly the following edge server and perform accordingly [1]. The proposed scheme takes these opportunities to transfer trust values of vehicles that leave the area of a particular edge server to the next edge server. We define three roles for an edge server: *current*, *transferor*, and *transferee*. The *current* edge server provides services to vehicles and calculates trust values of vehicles within its transmission range. The *transferor* edge server transfers information about vehicles, including current trust values, to the adjacent edge servers based on the vehicle's moving direction, route information, traffic condition, etc. Finally, *transferee* edge server receives information, provides services and updates trust values of vehicles [1]. The *transferee* edge server uses the received trust value of a vehicle as the basis for adjusting the trust value of the same vehicle while dwelling in its covered area.

An edge server can be compromised due to intrusions, physical attacks, malfunctions, node compromise, code injection attacks, etc. [32], [33]. A potential solution to defend malicious edge servers is to incorporate a centralized cloud server on top of the proposed edge-based vehicular model, where the cloud server monitors the behavior of an edge server based on the feedback of neighboring edge servers and its observation of edge behavior on deciding malicious and honest vehicles. If an edge server's behavior deviates significantly from the expected behavior, the cloud server decides an edge server is malicious and informs the neighbor edge servers and vehicles. In this case, the cloud server can temporarily take responsibility for computing trust values for vehicles belonging to the area of the malicious edge server. We plan to incorporate a concrete solution to handle this issue in our future work.

TABLE V
EVALUATION METRICS

| | |
|---|---|
| *Recall* | $\frac{NC}{\text{Total no. of malicious vehicles}}$ |
| *Precision* | $\frac{NC}{\text{No. of vehicles identified as malicious}}$ |
| *Accuracy* | $\frac{MC}{\text{Total no. of decisions}}$ |

NC=No. of correctly detected malicious vehicles
MC=No. of correct decision on vehicle type

## VI. EXPERIMENTAL EVALUATION

This section evaluates the performance of our proposed Fuzzy Logic-based Trust Estimation in Edge-enabled VANET (FLTEEV) scheme using a synthetic and a real-world public dataset. Its performance is compared against Decentralized Trust Calculation with Fuzzy Logic (DTCFL) scheme [10], and Novel Trust Calculation with Fuzzy Logic (NTCFL) scheme [11], already discussed in Section II.

### A. Evaluation Metrics

We used *Recall*, *Precision*, and *Accuracy* defined in Table V to evaluate the performance of the schemes. Four additional metrics are also considered to assess the performance in case of the synthetic dataset. They are:

*1. Packet Delivery Ratio* ($PDR_{del}$) shows the ratio between data packets received and the total packets sent from source to destination vehicles [11].

$$PRD_{del} = \frac{No.\ of\ data\ packets\ received}{No.\ of\ data\ packets\ sent}. \tag{4}$$

*2. Packet Dissemination Ratio* ($PDR_{dis}$) is the ratio between the accurate data packets received and the data packets generated at a source vehicle multiplied by the total number of vehicles [13].

$$PDR_{dis} = \frac{Packet_{true}}{Packet_{source} \times N} \tag{5}$$

where $\text{Packet}_{true}$ is the no. of packets received correctly, $\text{Packet}_{source}$ is the no. of packets generated by a vehicle, and $N$ is the total no. of vehicles.

*3. Messages per Data Packet* ($MDP$) is the average number of disseminated messages required to transmit information from source to destination vehicles [13].

$$MDP = \frac{No.\ of\ messages\ transmitted\ by\ all\ vehicles}{No.\ of\ data\ packets\ generated\ by\ sources}. \tag{6}$$

$MDP$ counts total no. of generated packets including the ACK and data packets.

*4. End-to-End (E2E) Delay* refers to the average time required for a data packet to be transmitted from the source to the destination vehicle [11]. Our experiment considered only the successfully delivered data packets to evaluate the E2E [13].

$$E2E = H \times D + T \times (N - 1). \tag{7}$$

TABLE VI
SIMULATION PARAMETERS

| Parameter | Value |
|---|---|
| Number of vehicles $N$ | 200 |
| Number of lanes | 4 (2 at each direction) |
| Channel fading model | Nakagami propagation model [10] |
| Vehicle speed | $36km/h \sim 150km/h$ [34] |
| Transmission range of a vehicle | $250m$ [6, 10] |
| Distance between RSUs | $1km$ [37] |
| Size of data packet | 512 bytes |
| Percentage of malicious vehicles $MV$ | 10%, 20%, 30%, 40% |
| Hop counts $H$ | $1 \sim 5$ |
| Simulation time | 500 *sec* [34] |

where $H$ is the number of hop counts, $D$ is the delay of first packet, $T$ is the transmission delay, and $N$ is the total number of data packets.

### B. Experiment With Synthetic Datasets

We simulated a freeway VANET [34] using OMNet++ ver. 5.6.1 [35] to conduct different experiments, where a freeway VANET is a controlled-access highway with two lanes in each direction. We used OMNet++ Mobility Framework [36] to construct the vehicular network, where OMNet++ simple modules (written in C++) represent vehicles and edge servers. Messages in the network were traveled through a chain of connections representing multi-hop communication. Vehicles (OMNet++ simple modules) were moved towards a straight line without changing their initial lane, and the edge servers (OMNet++ simple modules) were placed in specified positions maintaining the standard transmission range, which is 1 km [37]. Vehicles were assigned a unique identity and a velocity. They used data packets of 512 *bytes* to exchange traffic information where normal road condition was 0, and the congested or abnormal road condition was 1 to ease experiments. We can extend experiments for supporting correlation between multi-variable vectors of events [6]. Each data packet also maintains an integrity-protected vehicle list for multi-hop routing through relay vehicles [6]. Table VI presents the simulation parameters for our experiments.

In our simulation, we used the cSimpleModule class for packet transmission and analysis of received packets. We wrote customized C++ functions to compute factor values (FF, TF, MF, SF, and NTF) from the pre-defined fuzzy membership functions discussed in Section V-D.1. Besides, we wrote C++ functions on the specified edge servers' modules to determine the PDF, FPIF, and CAF from the fixed rule sets and fuzzy membership functions discussed in Sections V-D.2, V-D.3, and V-D.4. We then used the C++ functions to obtain the rank of vehicles from the fuzzy rule sets and compute trust values using the CoG method discussed in Sections V-E.1 and V-E.2.

We considered 200 vehicles for the experiment, where percentages of malicious vehicles were varied as 10%, 20%, 30%, and 40% as shown in Table VI. In each simulation, we specified the position and type of the vehicle (honest/malicious), where selected malicious vehicles arbitrarily drop, delay, modify, or inject data packets. Hostile vehicles executed *Packet Alteration Attacks* through content tampering, *Bad Mouth Attacks* by perfidiously adjusting trust factor values of vehicles with a probability of 0.3, and *On-Off Attacks* by dropping packets with a probability of 0.3 [10], [12]. The final results were the mean value of 30 simulations with 95% confidence intervals.

### C. Recall

Figures 5(a) $\sim$ 5(d) demonstrate the recall capability of different schemes for different $MV$ and $H$ when $N = 200$. The results reveal that the FLTEEV scheme is capable of detecting more hostile vehicles than the DTCFL and NTCFL schemes with higher values of $H$ and $MV$. The proposed scheme more accurately characterizes a malicious vehicle by analyzing every aspect of a hostile vehicle. It uses *PDF*, *CAF*, and *FPIF* to estimate packet forwarding ratio, content tampering, and false packet insertion ratio of a vehicle. The DTCFL scheme assesses the packet forwarding rate and content tampering using *FF* and *TF*, respectively. It does not consider the false packet injection and the impact of relay vehicles on message content. Therefore, the DTCFL scheme shows a mediocre performance in detecting mischievous vehicles. Besides, the NTCFL scheme partly considers the packet forwarding ratio using the *FF* in association with recommendation credibility *(RC)* and activity factor *(AF)*. Hence, the NTCFL scheme exhibits bad performance in detecting malicious vehicles. Besides, higher $MV$ and $H$ create complicated network scenarios by increasing the network size with newer paths affected by more malevolent vehicles. The FLTEEV handles this challenge more successfully visible from Figs. 5(a) $\sim$ 5(d). As the NTCFL puts the least effort into detecting rogue vehicles, its performance degrades severely with increasing $H$ and $MV$ as shown in Fig. 5(d).

Figure 5(e) demonstrates the performance of different schemes at $H = 5$. The FLTEEV scheme shows better recall than the DTCFL and NTCFL schemes for increasing $MV$. The FLTEEV, DTCFL, and NTCFL schemes detect nearly 88%, 76%, and 48% hostile vehicles, respectively, when $MV = 40\%$.

### D. Precision

Figures 6(a) $\sim$ 6(d) present the accuracy of different schemes in detecting hostile vehicles with increasing $H$ and $MV$ when $N = 200$. Among the mentioned schemes, the FLTEEV more exactly detects dishonest vehicles than the DTCFL and NTCFL schemes, as it considers all malevolent characteristics of vehicles in the trust estimation. Besides, Fig. 6(e) shows that when $H = 5$ and $MV = 40\%$, the NTCFL gains around 50%, the DTCFL achieves nearly 78%, and the FLTEEV attains approximately 88% precision. The result in Fig. 6(e) also demonstrates that the FLTEEV obtains approximately 13% and 76% improvement over the DTCFL and NTCFL schemes, respectively.

### E. Accuracy

Accuracy indicates the efficiency of decision-making that classifies vehicles according to their behavior.

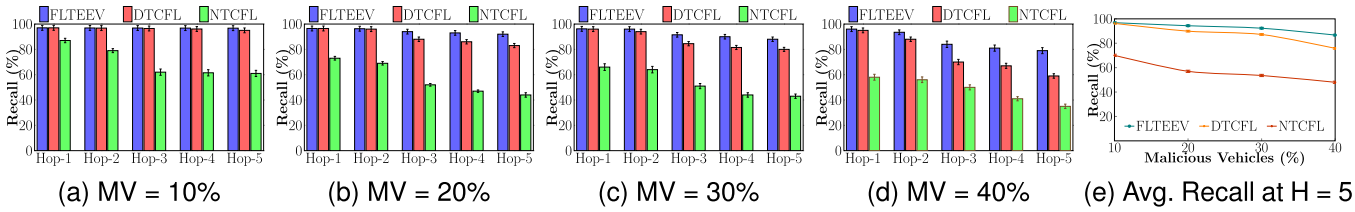


Fig. 5. Recall on various $H$ and $MV$ for $N$=200.

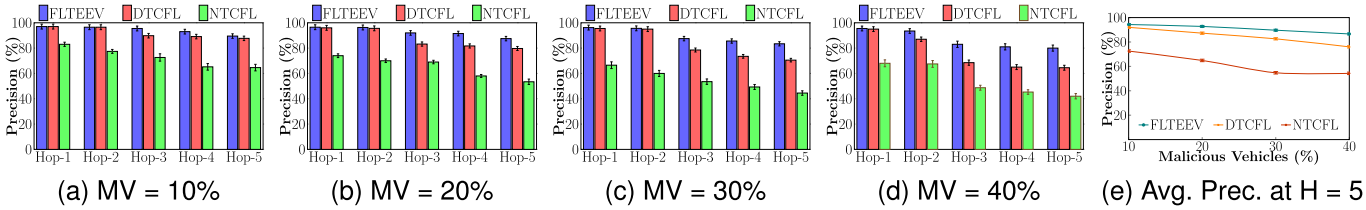


Fig. 6. Precision on various $H$ and $MV$ for $N$=200.

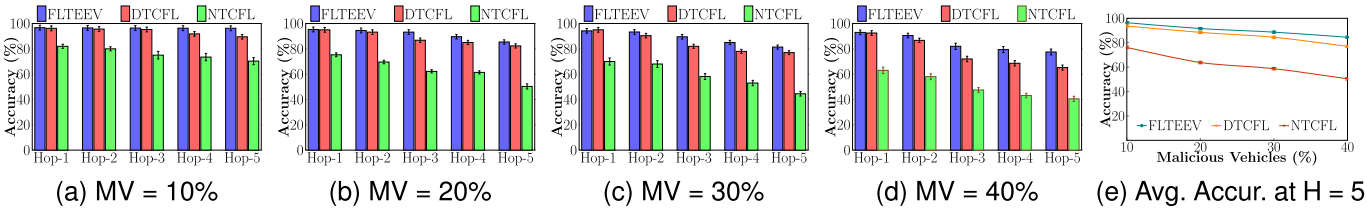


Fig. 7. Accuracy on various $H$ and $MV$ for $N$=200.

Figures. 7(a) $\sim$ 7(d) manifest that the FLTEEV shows better accuracy in identifying honest and malicious vehicles compared to the DTCFL and NTCFL schemes. Moreover, Fig. 7(e) indicates when $H = 5$ and $MV = 40\%$, the NTCFL, DTCFL, and FLTEEV schemes attain nearly 50%, 77%, and 85% accuracy in vehicle detection, respectively.

### *F. Packet Dissemination and Packet Delivery Ratio*

Packet dissemination ratio ($PDR_{dis}$) and packet delivery ratio ($PDR_{del}$) reflect the flow of data packets in networks. Figure 8(a) shows that the FLTEEV achieves a packet dissemination ratio of nearly 76% at $MV = 40\%$ whereas it is around 66% for the DTCFL and approximately 63% for the NTCFL. Similarly, the FLTEEV attains a higher delivery ratio (77%) compared to the DTCFL (72%) and the NTCFL (67%) at $MV = 40\%$, shown in Fig. 8(b). Due to a lack of accurate trust calculation, the DTCFL and NTCFL schemes cannot precisely identify trusted vehicles for packet transmission. Hence, packet drop, data alteration, and false packet injection by malicious vehicles lower the overall results shown in Figs. 8(a) and 8(b). The FLTEEV performs better on both metrics through precise trust computations, accurate detection of malicious vehicles, and assigning honest vehicles in packet transmission.

### *G. Messages per Data Packet and End-to-End Delay*

Messages per data packet (MDP) determines the number of messages transmitted to send information from source

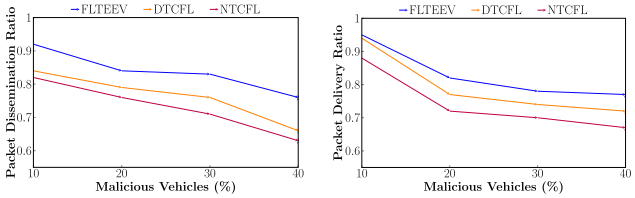


Fig. 8. Packet dissemination ratio and packet delivery ratio for various $MV$ when $N$=200 and $H = 5$.

to destination vehicles. Malicious relay vehicles drop or delay messages to hinder the normal flow of data packets, which causes packet re-transmissions by source vehicles. So detecting reliable relay vehicles is a key issue in handling effective data transmission in VANET. High values of MDP indicate the generation of redundant messages against the same data packet, which increases transmission time and end-to-end delay. Figure 9(a) indicates when $H = 5$ and $MV = 40\%$, the required MDP for the NTCFL, DTCFL, and FLTEEV schemes is around 22, 19, and 14, respectively. It depicts that the FLTEEV achieves approximately 37% and 27% improved performance than the NTCFL and DTCFL schemes, respectively. The FLTEEV performs better in the end-to-end (E2E) delay by accurately selecting honest relay vehicles to ensure data transmissions with minimum messages. Figure 9(b) indicates when $H = 5$ and $MV = 40\%$, the E2E delay for the NTCFL is around 1.5$s$, the DTCFL needs 1.3$s$, and the FLTEEV requires around 0.9$s$. The result indicates that the FLTEEV achieves approximately 40% and

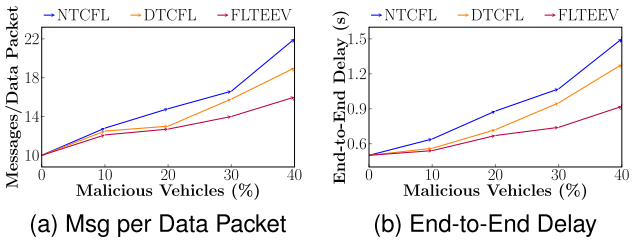


Fig. 9. Messages per data packet and end-to-end delay for various $MV$ when $N$=200 and $H = 5$.

31% less E2E delay than the NTCFL and DTCFL schemes, respectively.

### *H. Impact of the FLTEEV Scheme on VANET Routing Protocols*

We validated the competency of the FLTEEV scheme by incorporating it into two VANET proactive routing protocols: *Optimized Link State Routing Protocol (OLSR)* [38] and *Geographic Routing Protocol (GRP)* [38] and one reactive routing protocol: *Ad hoc On-demand Distance Vector (AODV)* [38]. Proactive routing protocols utilize the prior knowledge of network topology and awareness of a vehicle on its one-hop neighborhood to establish fixed and pre-defined routes [38]. On the contrary, reactive routing protocols perform the on-demand route discovery process and thus, require more time in data transmission [38].

*1) Packet Delivery Ratio:* Figure 10 shows that trust incorporation into routing protocols increases the packet delivery ratio compared to the basic protocols due to eliminating malicious vehicles from transmission paths. Figure 10 (a) depicts that OLSR with the FLTEEV trust model achieves approximately 20% more packet delivery ratio compared to the original OLSR protocol. Similarly, trust-assigned GRP and AODV attain around 22% and 21% improvement than the basic GRP and AODV, respectively, as shown in Fig. 10 (b) and 10 (c). Figure 10 (d) shows that OLSR achieves the highest packet delivery ratio, which AODV closely follows. Besides maintaining a global routing table for the entire network, OLSR also assures the availability of transmitted packets through predefined multi-point paths. While AODV selects on-demand relay vehicles according to trust values, GRP follows a greedy forwarding algorithm to select closest neighbors based on geographic position that can be inconsistent over time.

*2) End-to-End Delay:* Figure 11 compares the E2E delay of the fundamental OLSR, GRP, and AODV with the respective trust-assigned protocols where trust scores are calculated using the FLTEEV scheme. Trust helps to identify reliable relay vehicles in transmission paths, reducing overall delay by lessening the packet drop ratio. Figures 11(a) $\sim$ 11(c) indicate that the trust-assigned protocols show around 8%, 13%, and 18% efficiency compared to the respective basic OLSR, GRP, and AODV protocols, respectively. Figure 11(d) indicates that AODV has a higher E2E delay than GRP and OLSR. AODV executes the route discovery process before a packet transmission, while OLSR and GRP use predefined routes that are regularly updated according to network topology.

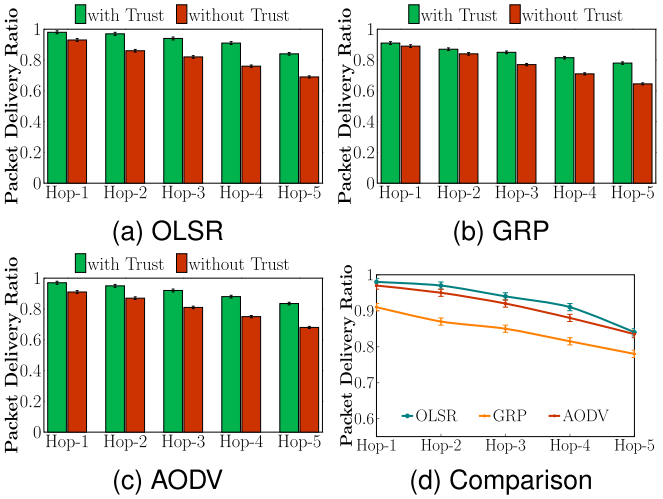


Fig. 10. Packet delivery ratio for routing protocols when $MV = 30\%$, $N = 200$.

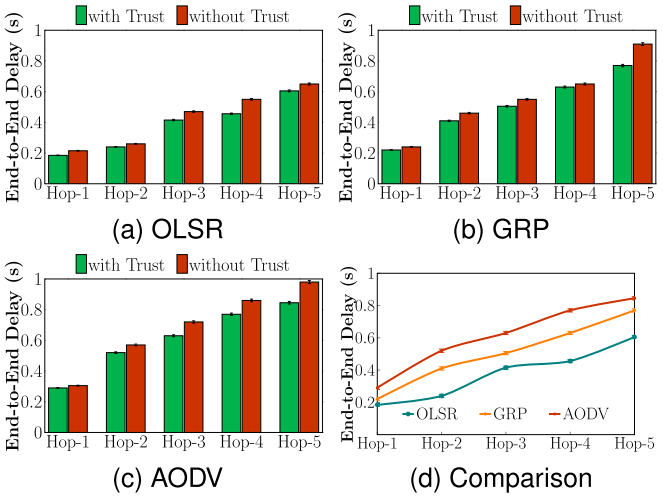


Fig. 11. End-to-end delay of routing protocols when $MV = 30\%$, $N = 200$.

*3) Throughput:* It indicates the amount of data received by vehicles per unit of time. Figure 12 shows the throughput of OLSR, GRP, and AODV protocols after the FLTEEV scheme is incorporated with them. Although each protocol exhibits good packet delivery ratio according to Fig. 10, throughput reduces for each protocol with higher values of $H$ shown in Fig. 12. This is due to the increased time required for packets to reach their destinations. With higher values of $H$, malicious vehicles also increase. These rogue vehicles initiate packet re-transmissions by dropping packets, which increases the time required for packets to reach the destination. Figure 12(d) depicts that the proactive protocols (OLSR, GRP) show a higher throughput ratio than the reactive protocol (AODV).

### *I. Experiment With Real-Valued Datasets*

In this experiment, we used the public dataset VeReMi [16] and its Extension [17]. Table VII shows the attack types ($AT$)

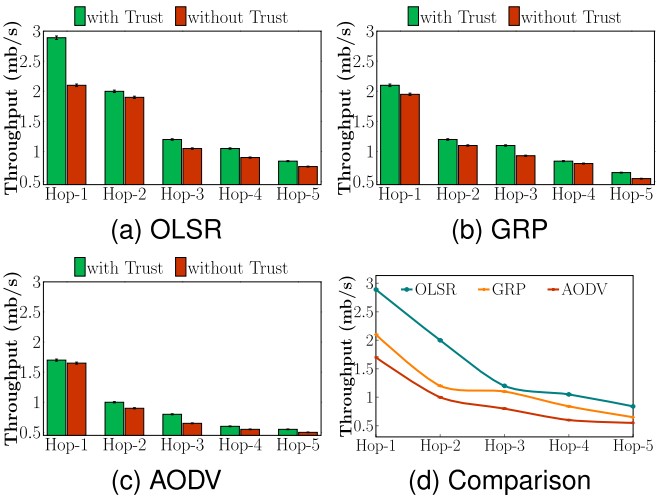


Fig. 12. Throughput of routing protocols when $MV = 30\%$ and $N = 200$.

TABLE VII
ATTACK DESCRIPTION OF VEREMI [16] AND VEREMI EXTENSION [17]

| Attack Type | Acronym | Description | Dataset |
|---|---|---|---|
| Constant position | CP | transmits fixed position | VeReMi & Ext. |
| Constant position offset | CPO | adds a fixed offset with actual position | VeReMi & Ext. |
| Random position | RP | shares random position | VeReMi & Ext. |
| Random position offset | RPO | adds random offset with exact position | VeReMi & Ext |
| Eventual stop | ES | vehicles behave normally for some time and then constantly transmit the last position | VeReMi & Ext. |
| Constant speed | CS | transmits fixed speed | VeReMi Ext. |
| Constant speed offset | CSO | adds a fixed offset with actual speed | VeReMi Ext. |
| Random speed | RS | shares random speed | VeReMi Ext. |
| Random speed offset | RSO | adds random offset with exact speed | VeReMi Ext. |
| Denial of Service | DoS | sends false packets to exhaust the capacity of a service provider | VeReMi Ext. |
| Data Replay | DR | replays previously captured data packets | VeReMi Ext. |

used in these datasets. Here, we only focus on the performance of the schemes on VeReMi dataset which considers five different attacks along with three attacker densities ($AD$) and three vehicle densities ($VD$).

Experimental results for different attacks in case of $AD = 30\%$ and different $VD$ are summarized in Table VIII. We computed the distance between vehicles separately using the position and speed values of consecutive packets, and used the inconsistency in distance values to identify false packets that ultimately detect malicious vehicles. The experimental results reveal that the FLTEEV achieves better recall, precision, and accuracy than the DTCFL and NTCFL schemes in all attack scenarios except the CPO attack. In this attack, distances derived from the position and speed values of consecutive packets cannot be distinguished, causing a problem in identifying malicious vehicles. Hence, both FLTEEV and DTCFL exhibit lower recall, precision, and accuracy than other attacks. On the other hand, both FLTEEV and DTCFL attain recall values of 1.00 for CP and RP attacks. In these attacks, the distance derived from position values of consecutive packets significantly vary from the distance obtained from the speed

TABLE VIII
RESULT ON VEREMI [16] DATASET

| $AT$ | $VD$ | Recall | | | Precision | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | FLTEEV | DTCFL | NTCFL | FLTEEV | DTCFL | NTCFL | FLTEEV | DTCFL | NTCFL |
| CP | L | **1.000** | **1.000** | 0.298 | **0.850** | 0.760 | 0.256 | **0.955** | 0.920 | 0.602 |
| | M | **1.000** | **1.000** | 0.368 | **0.750** | 0.630 | 0.292 | **0.904** | 0.831 | 0.558 |
| | H | **0.955** | **0.955** | 0.368 | **0.679** | 0.624 | 0.309 | **0.850** | 0.812 | 0.553 |
| CPO | L | 0.121 | 0.149 | **0.298** | **0.258** | 0.257 | 0.256 | **0.688** | 0.675 | 0.602 |
| | M | 0.319 | 0.310 | **0.368** | **0.311** | 0.295 | 0.292 | **0.597** | 0.585 | 0.558 |
| | H | **0.411** | 0.377 | 0.375 | **0.311** | 0.302 | 0.303 | **0.555** | 0.553 | 0.554 |
| RP | L | **1.000** | **1.000** | 0.300 | **0.861** | 0.747 | 0.262 | **0.958** | 0.913 | 0.600 |
| | M | **1.000** | **1.000** | 0.366 | **0.808** | 0.703 | 0.302 | **0.929** | 0.874 | 0.554 |
| | H | **0.994** | 0.993 | 0.371 | **0.754** | 0.641 | 0.295 | **0.903** | 0.835 | 0.559 |
| RPO | L | **0.877** | 0.839 | 0.319 | **0.868** | 0.679 | 0.266 | **0.937** | 0.863 | 0.616 |
| | M | **0.865** | 0.784 | 0.358 | **0.631** | 0.515 | 0.306 | **0.811** | 0.719 | 0.543 |
| | H | **0.839** | 0.760 | 0.381 | **0.510** | 0.432 | 0.296 | **0.746** | 0.675 | 0.560 |
| ES | L | **0.952** | **0.952** | 0.297 | **0.799** | 0.765 | 0.346 | **0.905** | 0.887 | 0.580 |
| | M | **0.983** | 0.974 | 0.373 | **0.561** | 0.556 | 0.281 | **0.781** | 0.777 | 0.560 |
| | H | **0.981** | 0.969 | 0.361 | **0.510** | 0.497 | 0.277 | **0.726** | 0.712 | 0.550 |

*L, M and H represent low, medium & high vehicle density, respectively.
**Best results in each simulation setup are highlighted in boldface.

values, leading to the identification of malicious vehicles. We observe that the performance gaps between the DTCFL and FLTEEV are comparatively narrow. There are mainly two reasons behind this behavior. Firstly, the VeReMi does not consider the uncertainty of VANET environments as it uses the two-ray signal interference model [16] (our synthetic dataset uses the Nakagami model [10] to consider the uncertainty in VANETs). Secondly, the VeReMi considers packet transmissions in the one-hop distance, which does not capture content alteration attacks by relay vehicles in multi-hop communications. Hence, the result analysis of the FLTEEV on the VeReMi dataset cannot utilize the advantage of the CAF. Besides, the FLTEEV cannot always take advantage of PDF, which correlates $SF$ with $FF$, and FPIF, which associates $TF$ with $MF$ (missing in the DTCFL scheme) due to not considering uncertainty in VeReMi. The FLTEEV shows approximately 9%, 2%, 8%, 13%, and 2% better accuracy in CP, CPO, RP, RPO, and ES attacks, respectively, compared to the DTCFL. Besides, the FLTEEV performs better than the DTCFL with increasing vehicle density. The NTCFL concentrates only on the packet forwarding ratio while paying no attention to the correctness of received data. Hence, this scheme shows almost constant performance for each $VD$ in different attack scenarios.

### J. Experiments With Extended VeReMi Dataset

The extended VeReMi dataset [17] includes a more accurate sensor error model, new attacks, and more data points. The misbehavior models are classified as malfunctions (position and speed malfunctions) and attacks (DoS, data replay, eventual stop attacks). The former describes benign behaviors caused by malfunctioning of on-board units (OBU)s or vehicle sensors, whereas the latter represents malicious behaviors caused by vehicles purposely relaying misleading information. All simulations in this dataset considers $AD = 30\%$ and $VD$ in rush hour (07h–09h) and low traffic times (14h-16h).

Table IX shows the experimental results on the VeReMi extension dataset. Alike the VeReMi dataset, the FLTEEV scheme offers better recall on the VeReMi extension for

TABLE IX
RESULT ON VEREMI EXTENSION [17] DATASET

| $AT$ | $VD$ | Recall | | | Precision | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | FLTEEV | DTCFL | NTCFL | FLTEEV | DTCFL | NTCFL | FLTEEV | DTCFL | NTCFL |
| CP | H | **1.000** | **1.000** | 0.347 | **0.881** | 0.723 | 0.296 | **0.960** | 0.887 | 0.560 |
| | L | **1.000** | **1.000** | 0.366 | **0.896** | 0.705 | 0.300 | **0.965** | 0.873 | 0.545 |
| CPO | H | 0.211 | 0.172 | **0.344** | **0.294** | **0.294** | **0.294** | **0.633** | 0.617 | 0.562 |
| | L | 0.233 | 0.175 | **0.368** | **0.306** | 0.300 | 0.300 | **0.627** | 0.600 | 0.545 |
| RP | H | **1.000** | **1.000** | 0.344 | **0.708** | 0.664 | 0.293 | **0.878** | 0.851 | 0.563 |
| | L | **1.000** | **1.000** | 0.368 | **0.716** | 0.647 | 0.300 | **0.879** | 0.834 | 0.545 |
| RPO | H | 0.992 | **0.995** | 0.344 | **0.702** | 0.661 | 0.294 | **0.874** | 0.848 | 0.562 |
| | L | 0.994 | **0.996** | 0.368 | **0.714** | 0.646 | 0.300 | **0.877** | 0.832 | 0.545 |
| CS | H | **0.963** | 0.875 | 0.344 | **0.877** | 0.695 | 0.294 | **0.949** | 0.850 | 0.562 |
| | L | **0.962** | 0.876 | 0.368 | **0.891** | 0.677 | 0.300 | **0.953** | 0.835 | 0.545 |
| CSO | H | 0.210 | 0.174 | **0.344** | 0.292 | 0.292 | **0.294** | **0.632** | 0.617 | 0.562 |
| | L | 0.290 | 0.280 | **0.360** | **0.303** | 0.301 | **0.303** | **0.587** | 0.584 | 0.548 |
| RS | H | 0.992 | **0.996** | 0.344 | **0.783** | 0.689 | 0.294 | **0.916** | 0.866 | 0.562 |
| | L | 0.992 | **0.997** | 0.368 | **0.792** | 0.669 | 0.300 | **0.918** | 0.848 | 0.545 |
| RSO | H | **0.983** | 0.941 | 0.344 | **0.705** | 0.651 | 0.294 | **0.873** | 0.833 | 0.562 |
| | L | **0.985** | 0.953 | 0.368 | **0.712** | 0.636 | 0.300 | **0.874** | 0.819 | 0.545 |
| ES | H | **0.859** | 0.828 | 0.344 | **0.633** | 0.597 | 0.294 | **0.811** | 0.784 | 0.562 |
| | L | **0.871** | 0.838 | 0.362 | **0.687** | 0.606 | 0.296 | **0.839** | 0.784 | 0.548 |
| DR | H | **0.975** | 0.828 | 0.344 | **0.693** | 0.612 | 0.294 | **0.866** | 0.795 | 0.563 |
| | L | **0.967** | 0.743 | 0.365 | **0.709** | 0.577 | 0.297 | **0.869** | 0.755 | 0.548 |
| DoS | H | **0.994** | 0.748 | 0.230 | **0.714** | 0.509 | 0.157 | **0.879** | 0.708 | 0.401 |
| | L | **0.994** | 0.732 | 0.253 | **0.714** | 0.493 | 0.170 | **0.875** | 0.685 | 0.388 |

*H and L represent high & low vehicle density, respectively.
**Best results in each simulation setup are highlighted in boldface.

different malfunction models, excluding CPO and CSO malfunctions, as they cannot accurately differentiate the misbehaving and honest vehicles. Once again, for CP and RP malfunctions, the FLTEEV and DTCFL achieve a recall value of 1.00 for the same reason discussed in Section VI-I. We observed that for RPO and RS, the DTCFL shows slightly better recall than the FLTEEV. This happens as the VeReMi does not consider uncertainty. In the cases of speed malfunctions, data falsification in the velocity property of a packet is challenging to detect compared to the falsification in position value, leading to a lower attacker detection for the speed-based malfunctions compared to the position-based malfunctions for both the FLTEEV and DTCFL. The FLTEEV shows around 11%, 5%, 6%, 6%, 14%, 3%, 8%, and 7% better accuracy in decision making than the DTCFL for the CP, CPO, RP, RPO, CS, CSO, RS, RSO, respectively.

For DoS attacks, each vehicle sends duplicate packets with the same position and velocity information, leading to false packet identification. The FPIF of the FLTEEV captures false packet injections due to $TF < MF$. In the DTCFL, the DoS attacks impact mainly on $FF$ and $TF$. Due to the high $FF$, the possibility of a malicious vehicle being considered as 'honest' increases, leading to lower attacker detection. The FLTEEV detects nearly 36% more malicious vehicles than the DTCFL for DoS attacks. In DR attacks, a vehicle repeats previously captured messages randomly, which can be detected from the consecutive packets' position and velocity information. This affects the FPIF of the FLTEEV and $TF$, $MF$ of the DTCFL. Due to considering the FPIF, our scheme identifies around 31% more attackers than the DTCFL. For both the VeReMi and its extension, the FLTEEV achieves better precision than the DTCFL in all cases, which signifies that the FLTEEV generates fewer false positives than the DTCFL.

Table X summarizes different aspects of the NTCFL, DTCFL, and FLTEEV schemes. The computational complexity analysis of the three schemes is provided in the Appendix.

## VII. CONCLUSION AND FUTURE WORK

We have proposed a trust estimation scheme for an edge-enabled VANET that utilizes fuzzy logic to deal with the uncertain and unpredictable behavior of vehicles and wireless communication channels. The proposed scheme detects malicious vehicles more accurately using three trust factors–*PDF*, *FPIF*, and *CAF* captured by fuzzy sets. Besides, we have introduced a new parameter *NTF* to capture the message alteration effect on the trust calculations. We have incorporated an inter-edge trust transfer mechanism to eliminate inconsistency in the VANET operation. The experimental results on synthetic datasets show that the proposed scheme attains recall, precision, and accuracy in 85 $\sim$ 90% to detect malicious vehicles. Moreover, this scheme reduces end-to-end delay nearly by 40% and message per data by approximately 37% compared to the contemporary works. Experimental results on real datasets show that our scheme attains nearly 36% and 31% improvement in detecting vehicles causing DoS and data replay attacks, respectively. Besides, our scheme improves network performance in different VANET routing protocols due to accurate trust calculation. Our scheme is also lightweight on vehicles as it outsources computation to edge servers. Hence, we believe that the proposed scheme is an efficient solution for the dynamic-nature VANET for detecting malicious vehicles. In the future, we plan to include more trust factors to cover a broader range of security attacks more accurately. Besides, we want to incorporate a concrete mechanism to handle malicious edge servers computing trust scores for vehicles.

## APPENDIX

### COMPUTATIONAL COMPLEXITY ANALYSIS

Table XI shows the computational overhead of the FLTEEV, DTCFL, and NTCFL schemes on vehicles and edge servers. We use $m$, $n$, $e$, $l$, and $p$ to denote the total number of one-hop neighbors, number of packets received in $T_{time\_slot}$ from a vehicle, number of events, number of rules in a fuzzy rule-base, and number of ranks in output membership function, respectively. For comparison equivalency, each scheme uses a hop count of 5. The computational complexity analyses of the three schemes are discussed below:

- *The FLTEEV Scheme:* It uses edge servers and splits the trust estimation into two parts: 1) computation in vehicles and 2) computation in edge servers. The computational complexity analysis of vehicles is further divided into 1) factor values calculation and 2) NTF computation. A vehicle determines factor values of $m$ one-hop neighbor vehicles in $\mathcal{O}(mn)$, where the computation of $FF$, $MF$, $TF$, and $SF$ requires $\mathcal{O}(mn)$, $\mathcal{O}(em)$, $\mathcal{O}(mn)$, and $\mathcal{O}(m)$, respectively, for $m$ neighbors and $n \geq e$. Besides, the vehicle calculates the network topology based $NTF$ that incurs $\mathcal{O}(emn^3)$ overhead. Vehicles apply the max-flow min-cut approach

TABLE X
COMPARISON OF THE FLTEEV SCHEME WITH THE DTCFL AND NTCFL SCHEMES

| Scheme | Trust factors used | Message tampering | False message injection | Drop/delay packets |
|---|---|---|---|---|
| DTCFL [10] | *FF*, *MF* and *TF* | Inaccurate | No | Inaccurate |
| NTCFL [11] | *FF*, *RC* and *AF* | No | No | Inaccurate |
| FLTEEV | *PDF*, *FPIF* and *CAF* | Accurate, using *CAF* | Yes, using *FPIF* | Accurate, using *PDF* |

TABLE XI
COMPUTATIONAL OVERHEAD COMPARISON

| Scheme | Individual vehicle | Edge server (per vehicle) |
|---|---|---|
| NTCFL [11] | $\mathcal{O}(mn) + \mathcal{O}(ml) + \mathcal{O}(mp)$ | — |
| DTCFL [10] | $\mathcal{O}(mn) + \mathcal{O}(ml) + \mathcal{O}(mp) + \mathcal{O}(m^6)$ | — |
| FLTEEV | $\mathcal{O}(mn) + \mathcal{O}(emn^3)$ | $\mathcal{O}(m)$+$\mathcal{O}(l)$+$\mathcal{O}(p)$ |

in the $NTF$ calculation that has a complexity of $\mathcal{O}(VE^2)$ where $V$ is the number of vertices, and $E$ is the number of edges in the network graph [39]. We assume that a vehicle computing NTF receives $n$ packets from individual one-hop neighbors. We also assume that the number of paths traveled by the $n$ packets is also $n$, where each path has a length of 5 hop counts. Hence, a packet generated from a source vehicle located at a 5 hop distance travels through 5 relay vehicles and 4 connecting edges. Thus, $n$ packets acquired from at most $n$ paths come across $5n$ relay vehicles and $(5-1)n$ connecting edges. Hence, the number of vertices in the topology graph is $5n$, and the number of edges is $(5-1)n$. We replace $V$ and $E^2$ with $n$ and $n^2$, respectively. The complexity of topology-based $NTF$ calculation for each event is $\mathcal{O}(n^3)$, which is $\mathcal{O}(emn^3)$ for $e$ events and $m$ one-hop neighbor vehicles. Thus, the overall computational overhead on each vehicle is $\mathcal{O}(mn) + \mathcal{O}(emn^3)$.

The edge server receives $m$ copies of $FF$, $MF$, $TF$, $SF$, and $NTF$ for each vehicle and calculates average value with a complexity of $\mathcal{O}(m)$. The edge server then computes $PDF$ and $FPIF$ for each vehicle by applying a fuzzy rule-base, and the complexity for this operation is $\mathcal{O}(l)$, where $l$ is the number of rules in the rule-base. The complexity of computing $CAF$ is $\mathcal{O}(1)$. After calculating $PDF$, $FPIF$, and $CAF$, the edge server applies the fuzzy rule-base approach to determine the rank of a vehicle with a complexity of $\mathcal{O}(l)$. It applies the rank of a vehicle on the output membership function and then computes the trust value of a vehicle using the Center of Gravity (COG) method following eq. (2) in the manuscript. As the number of output rank is $p$, according to eq. (2), there are $p$ median values $d_1, d_2, .., d_p$, and $p$ corresponding degree values $w_1, w_2, .., w_p$. Hence, the overall time to compute the trust value of a vehicle following eq. (2) is $\mathcal{O}(p)$. This leads to an overall overhead of $\mathcal{O}(m)$+ $\mathcal{O}(l)$+ $\mathcal{O}(p)$ on the edge server for each vehicle.

- *The NTCFL Scheme:* Trust calculation in the NTCFL is divided into two parts: 1) factor values calculation and 2) fuzzy rule-based trust calculation. Each vehicle calculates the forwarding factor, activity factor, and recommendation credibility for one-hop neighbor vehicles. The time complexity for factor values computation of $m$ neighbor vehicles is $\mathcal{O}(mn)$. The computational overhead of fuzzy rule-based trust calculation is $\mathcal{O}(ml)$, as each one-hop neighbor vehicle's factor values are compared against a fuzzy rule table comprising $l$ number of rules and $m$ is the number of one-hop neighbors. After determining the degree of trust level, a vehicle applies the COG method to get the trust value of a neighbor vehicle. Time complexity of COG method is $\mathcal{O}(p)$ which is $\mathcal{O}(mp)$ for $m$ one-hop neighbor vehicles. Therefore, the overall computational overhead on a vehicle in the NTCFL scheme is $\mathcal{O}(mn) + \mathcal{O}(ml) + \mathcal{O}(mp)$.
- *The DTCFL Scheme:* Trust calculation in the DTCFL comprises 1) factor values calculation, 2) fuzzy rule-based direct trust calculation, and 3) Q-learning based indirect trust calculation. Each vehicle calculates factor values for one-hop neighbors, where complexity of computing $FF$ is $\mathcal{O}(mn)$, $MF$ is $\mathcal{O}(em)$ and $TF$ is $\mathcal{O}(mn)$. The overall complexity of factor value calculation is $\mathcal{O}(mn)$ as $n \geq e$. The computational complexity of fuzzy rule-based direct trust calculation is $\mathcal{O}(ml)$+$\mathcal{O}(mp)$ where $\mathcal{O}(ml)$ is the time complexity of comparing factor values of $m$ neighbor vehicles against a fuzzy rule-base of $l$ rules and $\mathcal{O}(mp)$ is the computational overhead of computing the trust values of $m$ neighbors using the COG method (since the DTCFL does not mention the method of defuzzification, we use the COG method for comparison equivalency). According to [40], time complexity for reaching a goal state in Q-learning approach is $\mathcal{O}(N^3)$ in the worst-case situation, where $N$ indicates the size of the state space. As $m$ is the number of one-hop neighbors, maximum size of the state space for Q-learning is $N = m^2$. Thus, complexity of the Q-learning is $\mathcal{O}(m^6)$. Therefore, the overall complexity of the DTCFL is $\mathcal{O}(mn) + \mathcal{O}(ml) + \mathcal{O}(mp)$+$\mathcal{O}(m^6)$.

## ACKNOWLEDGMENT

The authors acknowledge the support of the University of Dhaka, Bangladesh in providing APC.

## REFERENCES

[1] G. G. M. N. Ali, P. H. J. Chong, S. K. Samantha, and E. Chan, "Efficient data dissemination in cooperative multi-RSU vehicular ad hoc networks (VANETs)," *J. Syst. Softw.*, vol. 117, pp. 508–527, Jul. 2016.

[2] M. Rath, B. Pati, and B. K. Pattanayak, "An overview on social networking: Design, issues, emerging trends, and security," in *Social Network Analytics: Computational Research Methods and Techniques*, vol. 21. Cambridge, MA, USA: Academic, 2018.

[3] C. An, C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, "A context-aware edge-based VANET communication scheme for ITS," *Sensors*, vol. 18, no. 7, p. 2022, Jun. 2018.

[4] Y. Lai et al., "Fog-based two-phase event monitoring and data gathering in vehicular sensor networks," *Sensors*, vol. 18, no. 2, p. 82, Dec. 2017.

[5] B. Al-Otaibi, N. Al-Nabhan, and Y. Tian, "Privacy-preserving vehicular rogue node detection scheme for fog computing," *Sensors*, vol. 19, no. 4, p. 965, Feb. 2019.

[6] J. Chen, G. Mao, C. Li, and D. Zhang, "A topological approach to secure message dissemination in vehicular networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 135–148, Jan. 2020.

[7] M. A. Talib, S. Abbas, Q. Nasir, and M. F. Mowakeh, "Systematic literature review on Internet-of-Vehicles communication security," *Int. J. Distrib. Sens. Netw.*, vol. 14, no. 12, Dec. 2018, Art. no. 1550147718815054.

[8] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "VANet security challenges and solutions: A survey," *Veh. Commun.*, vol. 7, pp. 7–20, Jan. 2017.
[9] S. A. Soleymani et al., "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.
[10] S. Guleng, C. Wu, X. Chen, X. Wang, T. Yoshinaga, and Y. Ji, "Decentralized trust evaluation in vehicular Internet of Things," *IEEE Access*, vol. 7, pp. 15980–15988, 2019.
[11] H. Xia, S.-S. Zhang, B.-X. Li, L. Li, and X.-G. Cheng, "Towards a novel trust-based multicast routing for VANETs," *Secur. Commun. Netw.*, vol. 2018, pp. 1–12, Oct. 2018.
[12] H. Xia, S.-S. Zhang, Y. Li, Z.-K. Pan, X. Peng, and X.-Z. Cheng, "An attack-resistant trust inference model for securing routing in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7108–7120, Jul. 2019.
[13] C. Wu, S. Ohzahata, and T. Kato, "VANET broadcast protocol based on fuzzy logic and lightweight retransmission mechanism," *IEICE Trans. Commun.*, vol. E95–B, no. 2, pp. 415–425, Feb. 2012.
[14] S. A. Soleymani et al., "A trust model using edge nodes and a cuckoo filter for securing VANET under the NLoS condition," *Symmetry*, vol. 12, no. 4, p. 609, Apr. 2020.
[15] Md. M. Hasan, M. Jahan, S. Kabir, and C. Wagner, "A fuzzy logic-based trust estimation in edge-enabled vehicular ad hoc networks," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2021, pp. 1–8.
[16] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Proc. 14th EAI. Int. Secur. Privacy Commun. Netw. (SecureComm)*, Dec. 2018, pp. 318–337.
[17] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi extension: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
[18] A. Hesham, A. Abdel-Hamid, and M. A. El-Nasr, "A dynamic key distribution protocol for PKI-based VANETs," in *Proc. IFIP Wireless Days (WD)*, Oct. 2011, pp. 1–3.
[19] L. He and W. T. Zhu, "Mitigating DoS attacks against signature-based authentication in VANETs," in *Proc. IEEE Int. Conf. Comput. Sci. Autom. Eng. (CSAE)*, vol. 3, May 2012, pp. 261–265.
[20] I. A. Sumra, H. Hasbullah, J. Lail, and M. Rehman, "Trust and trusted computing in VANET," *Comput. Sci. J.*, vol. 1, no. 1, pp. 1–24, Aug. 2011.
[21] Z. Liu, J. Ma, Z. Jiang, H. Zhu, and Y. Miao, "LSOT: A lightweight self-organized trust model in VANETs," *Mobile Inf. Syst.*, vol. 2016, no. 2, pp. 1–15, Jan. 2016.
[22] X. Zhang, C. Lyu, Z. Shi, D. Li, N. N. Xiong, and C.-H. Chi, "Reliable multiservice delivery in fog-enabled VANETs: Integrated misbehavior detection and tolerance," *IEEE Access*, vol. 7, pp. 95762–95778, 2019.
[23] C. Wu, T. Yoshinaga, X. Chen, L. Zhang, and Y. Ji, "Cluster-based content distribution integrating LTE and IEEE 802.11p with fuzzy logic and Q-learning," *IEEE Comput. Intell. Mag.*, vol. 13, no. 1, pp. 41–50, Feb. 2018.
[24] A. Mahmood, B. Butler, W. E. Zhang, Q. Z. Sheng, and S. A. Siddiqui, "A hybrid trust management heuristic for VANETs," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2019, pp. 748–752.
[25] J. Zhang, K. Zheng, D. Zhang, and B. Yan, "AATMS: An anti-attack trust management scheme in VANET," *IEEE Access*, vol. 8, pp. 21077–21090, 2020.
[26] M. G. Al Zamil et al., "False-alarm detection in the fog-based Internet of Connected Vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7035–7044, Jul. 2019.
[27] J. Radak, B. Ducourthial, V. Cherfaoui, and S. Bonnet, "Detecting road events using distributed data fusion: Experimental evaluation for the icy roads case," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 184–194, Jan. 2016.
[28] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Apr. 2008, pp. 1238–1246.
[29] T. Bouali, S.-M. Senouci, and H. Sedjelmaci, "A distributed detection and prevention scheme from malicious nodes in vehicular networks," *Int. J. Commun. Syst.*, vol. 29, no. 10, pp. 1683–1704, Jul. 2016.
[30] K. F. Hasan, C. Wang, Y. Feng, and Y.-C. Tian, "Time synchronization in vehicular ad-hoc networks: A survey on theory and practice," *Veh. Commun.*, vol. 14, pp. 39–51, Oct. 2018.
[31] I. Okhrin and K. Richter, "Vehicle routing problem with real-time travel times," *Int. J. Veh. Inf. Commun. Syst.*, vol. 2, nos. 1–2, pp. 59–77, 2009.
[32] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[33] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 779–811, 1st Quart., 2019.
[34] L. A. Hassnawi, R. B. Ahmad, A. Yahya, H. R. Alsanad, and S. A. Aljunid, "Impact of freeway mobility pattern on routing performance of mobile ad hoc networks," *J. Next Gener. Inf. Technol.*, vol. 4, no. 3, pp. 139–150, May 2013.
[35] A. Varga, "Using the OMNeT++ discrete event simulation system in education," *IEEE Trans. Educ.*, vol. 42, no. 4, p. 11, Nov. 1999.
[36] M. Löbbers and D. Willkomm, "A mobility framework for OMNeT++ user manual," Version 1.0a4, Telecommunication Networks Group-Technische Universitaet, Berlin, Germany, 2007.
[37] H. Zhao, Y. Zhu, J. Tang, Z. Han, and G. S. Aujla, "Message-sensing classified transmission scheme based on mobile edge computing in the Internet of Vehicles," *Softw., Pract. Exper.*, vol. 51, no. 12, pp. 2501–2518, Dec. 2021.
[38] A. Yasser, M. Zorkany, and N. A. Kader, "VANET routing protocol for V2V implementation: A suitable solution for developing countries," *Cogent Eng.*, vol. 4, no. 1, Jan. 2017, Art. no. 1362802.
[39] T. H. Coremen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
[40] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning," in *Proc. AAAI*, 1993, pp. 99–107.

**Md. Mahmudul Hasan** received the bachelor's and master's degrees in computer science and engineering from the University of Dhaka, Bangladesh. His current research interests include VANET security, web security, and network security.

**Mosarrat Jahan** (Member, IEEE) received the bachelor's and master's degrees in computer science and engineering from the University of Dhaka, Bangladesh, the master's degree in computer science (research) from Concordia University, Canada, in 2012, and the Ph.D. degree from the School of Computer Science and Engineering, UNSW Australia. Currently, she is an Associate Professor with the Department of Computer Science and Engineering, University of Dhaka. Her current research interests include VANET security, cloud security, the IoT security, blockchain, and applied cryptography.

**Shaily Kabir** (Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in computer science and engineering (CSE) from the University of Dhaka, Bangladesh, the master's degree in computer science from Concordia University, Canada, in 2012, and the Ph.D. degree in computer science from the University of Nottingham, U.K., in 2020. Currently, she is an Associate Professor of CSE with the University of Dhaka. Her current research interests include machine learning, AI, uncertainty modeling, and data science.