BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0042

DoSRT: A Denial-of-Service Resistant Trust Model for VANET

Niharika Keshari, Dinesh Singh, Ashish Kumar Maurya

CSED, MNNIT Allahabad, Prayagraj, India
E-mails: niharika@mnnit.ac.in dinesh_singh@mnnit.ac.in ashishmaurya@mnnit.ac.in

Abstract: *The Denial of Service* (*DoS*) *attack threatens the availability of key components of Vehicular Ad-hoc Network* (*VANET*)*. Various centralized and decentralized trust-based approaches have been proposed to secure the VANET from DoS attack. The centralized approach is less efficient because the attack on the central trust manager leads to the overall failure of services. In comparison, the cluster-based decentralized approach faces overhead because of frequent changes in cluster members due to the high speed of the vehicles. Therefore, we have proposed a cluster-based Denial-of-Service Resistant Trust model* (*DoSRT*)*. It improves decentralized trust management using speed deviation-based clustering and detects DoS attack based on the frequency of messages sent. Through performance evaluation, we have found that DoSRT improves precision, recall, accuracy, and F-Score by around 19%, 16%, 20%, and 17% in the presence of 30% DoS attackers.*

Keywords: *Vehicular Ad-hoc Network* (*VANET*)*, Trust management, DoS attack, Denial-of-Service Resistant Trust model* (*DoSRT*)*, Dedicated short-range communication* (*DSRC*)*.*

1. Introduction

With the rapid advancement in wireless communication and the automobile industry, Vehicular Ad-hoc Networks (VANET) have attracted researchers' attention. Vehicles are being equipped with smart devices such as a Global Positioning System (GPS), sensor, and other service-providing devices that help to share information by forming a spontaneous network known as VANET [1]. The establishment of a VANET-based Intelligent Transportation System (ITS) emerged in the United States Congress through the demonstration of the Transportation Efficiency Act of 1991 [2]. This information is shared through two modes of communication (as shown in Fig. 1): Vehicle-to-Vehicle (V2V) [3], in which the vehicle communicates directly with each other, and Vehicle-to-Infrastructure (V2I) [4], in which the vehicle communicates with the roadside infrastructure called Roadside Units (RSUs) [5]. These entities interact together to provide safety, alert, emergency, infotainment information, etc., to road users for safe and comfortable driving [6].

The life of the user is one of the important and critical factors, which depends on the security of VANET. The security of VANET is vulnerable if any of the fundamental security services (availability, confidentiality, integrity, non-repudiation) is threatened by the attacker. All these attacks catastrophically affect the system, and among those DoS attack is the prime one. DoS attack affects the availability of the system and tends the system lead toward partial or total interruption and cause loss of vital information.

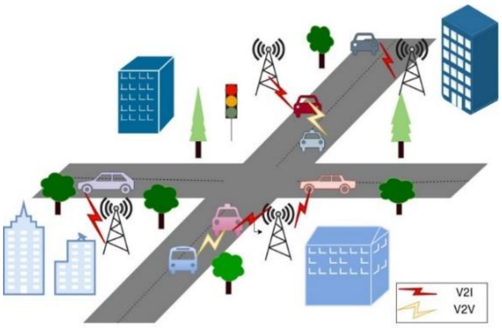

Fig. 1. Communication in VANET

To overcome the above threat, researchers have explored various security solutions based on certificates, digital signatures, Public Key Infrastructure (PKI), etc. The traditional solution given can prevent the system from an outside attacker but is unable to deal with authentic inside attackers [7]. Hence, to secure the system from inside attackers, a trust model has been proposed. The motivation behind designing the trust model for VANET is to resist the network against internal attackers from spreading/disseminating inaccurate, unauthentic, and forged messages among the entities. However, due to the high mobility of vehicles, evaluation and estimation of trust for received message or sender vehicle is a challenge. Trust value defines the belief of the trustor over the trustee. It allows us to determine whether the vehicle is malicious or not (according to the time and situation). Also, trust evaluation (for the sender or received information) in VANET becomes challenging because of dynamic topology, varying numbers of malicious vehicles, and the absence of monitoring through a trusted third party [8]. Moreover, during trust establishment, each vehicle evaluates the trust of its neighboring vehicle and, after that, informs the misbehavior detection system through centralized access points like RSUs. Hence, to generate the final trust report, the central server has to communicate with each vehicle to integrate the evaluated trust. It increases the workload on the centralized server as well as delays in gaining the trust report. Another problem is if, due to any circumstance, infrastructure is unavailable, then the availability of service is also affected. Therefore, a centralized system for trust evaluation is not suitable [9, 10]. On the other hand, the cluster-based approach overcomes the above problem by integrating the trust report of cluster members through the cluster head. Cluster formation reduces the communication between vehicles and infrastructure. However, the dynamic change in topology increases cluster switches very frequently.

Based on the above problem, we have been motivated to propose a trust model based on clustering that not only evaluates the trust but selects a stable clustering environment to reduce overhead due to cluster switch.

The proposed model calculates the trust of the vehicles based on the frequency of message exchange to resist DoS attacks. The contributions of the proposed approach as summarized below:

1. We proposed a DoS-resistant trust framework to stop the untrustworthy vehicle from tempering resource availability.

2. Our proposed DoSRT model incorporates two evaluation parameters: Direct trust and Indirect Trust. Both of the parameters are evaluated through the behavioral pattern of the vehicle. The collective use of these parameters helps to eliminate the problem of inside attacks.

This paper is organized as follows. Section 2 gives an overview of some existing related work. The details of our proposed DoSRT model are described in Section 3. Section 4 presents simulation result and analysis. Finally, the conclusion and future research direction are given in Section 5.

2. Related work

In this section, we survey the techniques, merits, and limitations of some existing trust evaluation methods in VANET. Some conventional methods are based on the direct calculation [11] of trust based on some pre-defined communication attribute between vehicles (sender and receiver). In comparison, other uses indirect trust calculation [12] based on the neighbor’s feedback/recommendation. Nevertheless, trust evaluation only through recommendation may lead to a collision attack in which a secret agreement through an adversary may illus the trustworthy vehicle as untrustworthy. Hence, combined evaluation (both direct/indirect trust) can overcome the above problem. In the literature, we have found another class of category of trust management method (at access points like RSUs): decentralized among the participant vehicles centralized within the infrastructure, or a hybrid of centralized/decentralized. In a decentralized approach, each vehicle evaluates the trust value of any other (which means a trust computation module is installed on every vehicle), while in a centralized approach, a central authority will monitor all communication, analyze and maintain historical data, and compute the trust of the vehicle or information. A centralized scheme suffers from additional overhead and increased detection time. In addition, the resulting trust model is divided into two ways, i.e., (1) node-centric trust model, and (2) data-centric trust model.

2.1. Node-centric trust model

The node-centric trust model deals with the trustworthiness of the sender vehicle rather than the information sent by it. The main aim of the node-centric trust model is to identify malicious vehicles among legitimate vehicles. Besides, node trust is also a key measure for the provision of secure routing for the efficient delivery of VANET data. Chen and Wang [14] have proposed a decentralized cloud-based trust model where Vehicular Cloud Network (VCN) architecture computes the trust of a vehicle on the basis of past direct node interactions, friends' behavior, and neighbor's behavior (attributes as previous trust value trusted friend list, direct neighbor list, and unknown neighbor list). The proposed approach reduces the average response time during trust calculation using the cloud, and the drawback of this trust model is the scheduling of resources to the vehicles at the vehicular cloud layer.

Another cloud-based interaction-time prediction algorithm has been designed [15] to resolve instability using internal and external similarities among vehicles. The problem with this approach is extra overhead due to multiple dynamic sources to estimate the trust value of the sender.

An approach discussed by Ning et al. [16] detects traffic anomalies using trajectory data analysis. The designed framework models the traffic and detects the anomalies. However, this approach requires a huge amount of data, powerful computational ability, and massive storage capacity.

Similarly, a cluster-based reputation trust model is presented in [17, 18], where each node cooperates with others to prevent unilateral decisions and aggregate judgments to improve stability, overhead, and delay. El Khatib et al. [19] aggregate judgment using a watchdog and Artificial Neural Network (ANN), which results in a collision attack when multiple watchdog nodes collude together to impersonate a trustworthy vehicle. To overcome the above problem, research work in [20] presents an approach by using watchdog (to observe multi-relay node) and Dempster Shafer Theory (DST) (to make the final decision from evidence). Reward and punishment are given based on the aggregation of final decisions and cooperative behavior. The absence of learning and a highly dynamic environment is an important constraint of the work of [20].

Due to the high dynamic of VANET, the system leads towards failure while collecting ample data about the neighbors. The static node-centered trust evaluation methods are unsuitable for VANET because of their application-based nature and in-variance in time or slow-evolving parameters.

2.2. Data-centric trust model

The data-centric trust model indicates the truthfulness of data rather than the sender of the data. Data-centric trust is usually referred to as the event-centric trust model, which primarily focuses on determining the accuracy or reliability of the data and detecting invalid or false data in VANETs.

For instance, a decentralized agent-based data-centric trust model has been proposed by Minhas et al. [21] to resist VANET from the dissemination of false messages from malicious nodes. The trust model deals with data sparsity, scalability, and privacy preservation. The key issue with the model is that it takes time to determine and distribute trusted neighborhood information in a real-time environment. Another data-centric approach has been designed by Fogue et al. [22] to reduce the latency of the event message dissemination process using position verification. Trusted, safe, and prominent data, together with their freshness and location relevance, are very important and useful for traffic in VANET [23]. The accuracy of determining the trustworthy vehicle is drastically reduced if the vehicle advertises incorrect information about its position.

Due to the inherent dynamic nature of VANET, current data-centered trust evaluation methods are concentrated on time proximity, location proximity, and number of reports for the same incident or different types of incident. Raya and

Hubaux [24], propose a data-centric trust model based on the aggregation of multiple reports for incidents (because it is difficult to determine the trustworthiness of information based on a single report). Here, the distance between the event and the vehicle, the number of sensors deployed to detect the same event, the vehicle's detection range, and the weight of the vehicle (according to the vehicle type) are used to determine the trustworthiness of an alert message.

The author of [25] has proposed a cluster-based secure approach that considers forwarding delay, vehicle velocity, and a number of trusted confident neighbors as clustering metrics. The approach being designed is useful for the constant speed mobility model for the stability of the formed cluster but becomes useless when overspeeding and underspeeding occurs.

DoSRT overcomes the problem of overspeeding and underspeeding using a stable clustering approach designed by Rawashdeh and Mahmud [26] and resists VANET from DoS attackers. Our DoSRT model ensures resource availability by evaluating trust on the basis of the frequency of the received beacon.

3. Proposed Denial-of-Service Resistant Trust model (DoSRT)

In this section, we elaborate on DoSRT, in which each vehicle monitor is the behavior of its neighbor vehicle continuously at each received message. First, we give an abstract overview of the cluster-based trust model followed by cluster formation and trust evaluation. The architecture model given in Fig. 2 suggests various steps in order to evaluate the trust of the vehicle. Evaluation is carried out using two stages, including (shown in Fig. 3) direct trust computation and indirect trust computation. First, it evaluates the direct trust of the sender vehicle according to the frequency of beacon transfer.

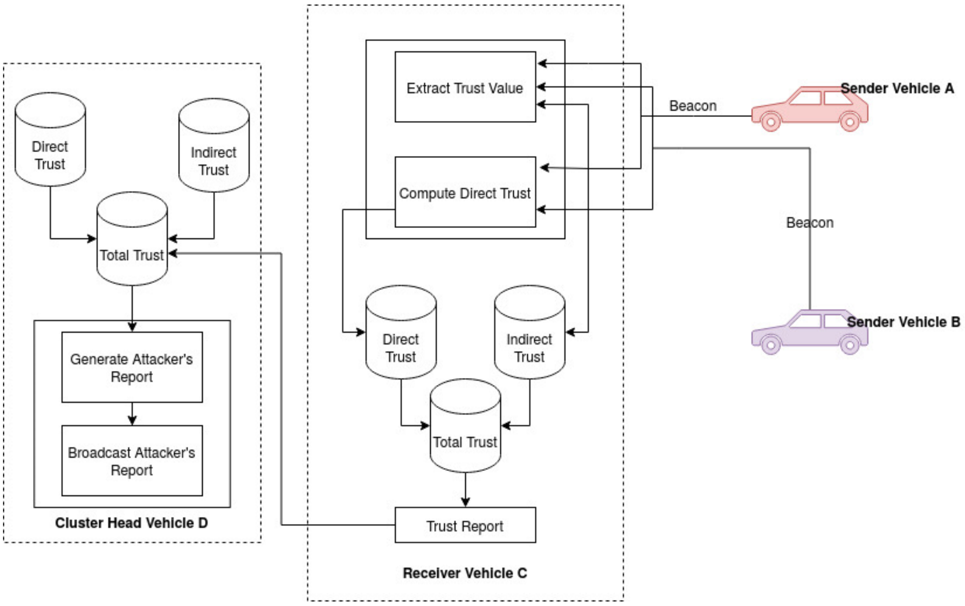

Fig. 2. Architecture model for trust evaluation

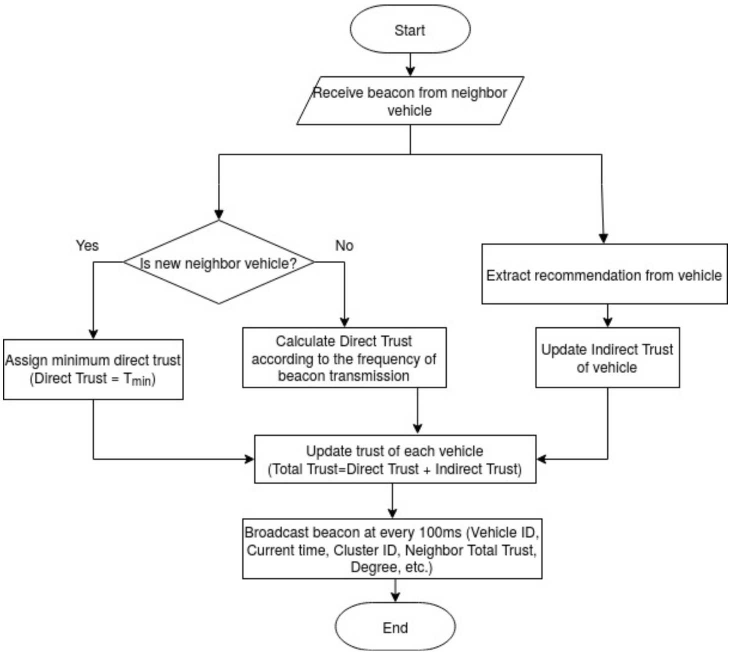
Start
Receive beacon from neighbor vehicle
Is new neighbor vehicle?
Yes
No
Extract recommendation from vehicle
Assign minimum direct trust (Direct Trust = T_min)
Calculate Direct Trust according to the frequency of beacon transmission
Update Indirect Trust of vehicle
Update trust of each vehicle (Total Trust=Direct Trust + Indirect Trust)
Broadcast beacon at every 100ms (Vehicle ID, Current time, Cluster ID, Neighbor Total Trust, Degree, etc.)
End

Fig. 3. Flow chart of trust calculation

The next equation is used to determine how frequently the beacon is transferred by the sender vehicle:

(1) $$T_{\text{direct}}(r,s) = \frac{T_S(t)-T_S(t-1)}{100},$$

where, $T_{\text{direct}}(r, s)$ is the direct trust of sender s evaluated by receiver r, and $T(t)$ is the Message received at t-th time.

Secondly, it calculates the indirect trust from the recommendation given by the neighbor as

(2) $$T_{\text{indirect}}(r,s) = (1-\alpha) * T_{\text{tt}}^{\text{rf}}(\text{rf},s) + \alpha * T_{\text{tt}}^{r}(r,s),$$

where, $T_{\text{indrect}}(r, s)$ is indirect trust of vehicle s calculated by vehicle r, $T_{\text{tt}}^{\text{rf}}(\text{rf},s)$ is recommendation of vehicle s is received recommend forwarder vehicle rf, $T_{\text{tt}}^{r}(r, s)$ is total trust of vehicle s calculated by vehicle r, and α is cosine similarity.

The cosine similarity finds the similarity between the total trust calculated by the receiver r (T_{tt}^{r}) and received recommendation T_r using

(3) $$\cos\alpha = \frac{T_{\text{tt}}^{r}.T_r}{|T_{\text{tt}}^{r}|.|T_r|}.$$

Further, it integrates both direct and indirect trust to provide higher accuracy and low overhead to detect the malicious vehicle the cluster head aggregates all trust reports of cluster members and broadcasts the malicious vehicle list as defined in the algorithm, as shown

(4) $$T_{\text{tt}}^{r}(r,s) = \frac{T_{\text{direct}}(r,\ s) \mp T_{\text{indirect}}(r,\ s)}{2}.$$

Each vehicle sends a beacon every 100 ms to notify its neighbor about the position, direction, degree, and vehicle type, as well as neighboring information (in which cluster it belongs and the trustworthiness of the neighborhood vehicle). The beacon frame format is shown in Fig. 4.

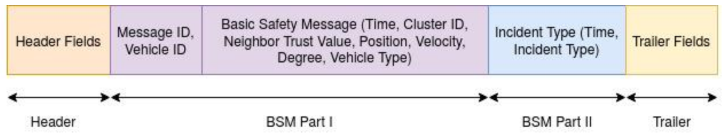
Header Fields
Message ID, Vehicle ID
Basic Safety Message (Time, Cluster ID, Neighbor Trust Value, Position, Velocity, Degree, Vehicle Type)
Incident Type (Time, Incident Type)
Trailer Fields
Header
BSM Part I
BSM Part II
Trailer

Fig. 4. Beacon frame format

3.1. Cluster formation

Clustering in VANET is one of the fundamental approaches used to control the topology change. Many of the VANET clustering methodologies in literature have emerged from the Mobile Ad hoc NETwork (MANET). However, VANET can be defined as a high-speed MANET, and the presence of VANET nodes in the same geographical area does not indicate that they have the same pattern of mobility. The degree of speed difference between neighbors should be considered, therefore by the clustering scheme in order to form a stable cluster. Our proposed model considers the stable clustering technique proposed in [26]. This technique is based on speed difference to create a relatively stable cluster with reduced cluster switch and increased cluster lifetime. Using this approach, our network is partitioned into a minimum number of clusters, where each member in the cluster has the same mobility pattern (speed difference ± 10 km/h as shown in Fig. 5 and 6). After cluster formation, each cluster member evaluates trust (discussed in the next subsection) to find malicious vehicles.

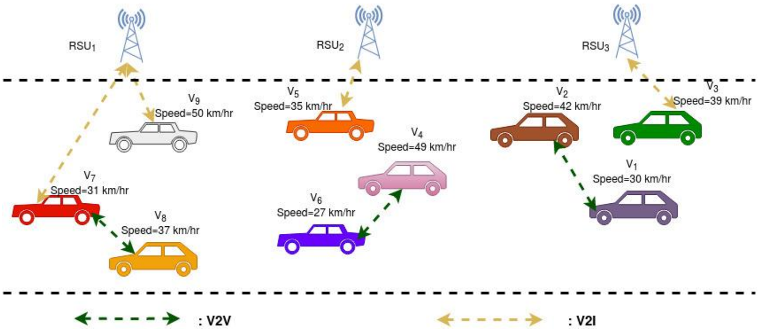

Fig. 5. Vehicles at time *t*

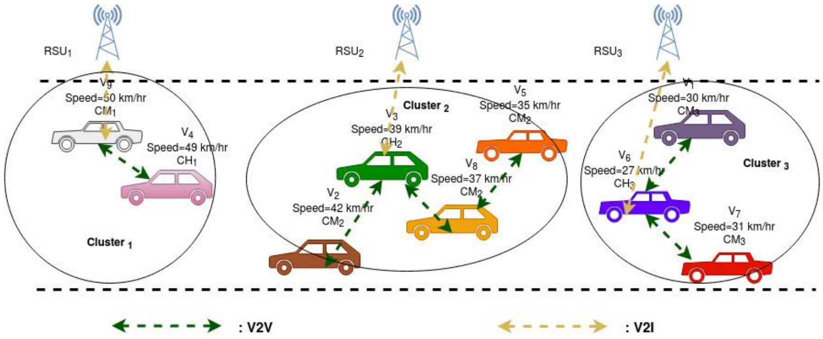
RSU$_1$
RSU$_2$
RSU$_3$
V$_9$ Speed=50 km/hr CM$_1$
V$_4$ Speed=49 km/hr CH$_1$
Cluster $_1$
Cluster $_2$
V$_3$ Speed=39 km/hr CH$_2$
V$_5$ Speed=35 km/hr CM$_2$
V$_8$ Speed=37 km/hr CM$_2$
V$_2$ Speed=42 km/hr CM$_2$
V$_1$ Speed=30 km/hr CM$_3$
V$_6$ Speed=27 km/hr CH$_3$
Cluster $_3$
V$_7$ Speed=31 km/hr CM$_3$
: V2V
: V2I

Fig. 6. Vehicles at time *t*+*t′*

3.2. Trust evaluation

DoSRT evaluates the trustworthiness (Table 1 shows the Description of the symbol used for evaluation, and trust evaluation is shown in Table 2) of the vehicles based on the receiving beacon frame.

Table 1. Symbol description

Acronym/Symbol	Description
V_s	Sender vehicle
V_r	Receiver vehicle
T_{direct}	Direct trust list
T_{indirect}	Indirect trust list
$T_{\text{direct}}(r, s)$	Direct trust value of V_s evaluated by V_r
$T_{\text{indirect}}(r, s)$	Indirect trust value of V_s evaluated by V_r
$T_s(t)$	Beacon received from V_s t milliseconds
$T_{\min}$	Minimum trust value
T_{tt}^{r}	A list contains the total trust values of V_r ‘s neighbor’s vehicle
$T_{\text{tt}}^{r}(r, s)$	The total trust list of V_s is evaluated by V_r
α	Cosine Similarity
T_{thresh}	Threshold value to detect malicious vehicle
n	Number of vehicles
L_a	Attacker list
$L_a(r, s)$	Vehicle V_r adds V_s to attacker list

Table 2. Trust evaluation

Vehicle ID	Cluster ID	Degree	Speed, Km/h	Neighboring vehicle trust
V_1	3	CM	30	V_5=7.1, V_6=8.9
V_2	2	CM	42	V_3=8, V_8=3.4
V_3	2	CH	39	V_2=7.2, V_8=4.6, V_5=2
V_4	1	CH	49	V_9=6.3
V_5	2	CM	35	V_1=8.3, V_6=7.9, V_8=5.0, V_3=7.1
V_6	3	CH	27	V_1=8, V_5=6.7, V_5 =8.1
V_7	3	CM	31	V_6=8
V_8	2	CM	37	V_5=4.3, V_3=8.5, V_2=7.9
V_9	1	CM	50	V_4=5.9

There are four modules in DoSRT, i.e., (1) Direct-trust computation of sender, (2) Indirect-trust computation, (3) Total trust computation, and (4) Generate attacker list. The detailed discussion of DoSRT is explained as follows.

- **Direct trust computation of sender.** The beacon consists of vehicle id, speed, position, neighbor’s trust value, degree, and time stamp at the time of creation. When a vehicle receives a beacon for the first time, it assigns direct trust as the initial minimum trust value ($T_{\min}$ = 0.5) and saves the time at which the beacon has been received. $T_{\min}$ defines that the behavior of a vehicle is neither trustworthy nor untrustworthy. On the other side, if the vehicle has received a beacon previously, it evaluates the direct trust on the basis of the rate of beacon transfer. At this point, it finds the time difference of the currently received beacon with respect to the last received beacon and divides it with the actual beacon transfer rate (which is 100 ms).

Algorithm 1 shows the procedure followed for direct trust calculation. Step 1 shows the beacon received by receiver V_r from V_s. Steps 2-3 are direct trust calculation by vehicle V_r when the vehicle is not a new neighbor. While adding a new neighbor vehicle and assigning minimum trust is presented in Steps 4-6.

Algorithm 1. Direct Trust Computation

```
Input: $M$, $T_s(i)$
Output: $T_{\text{direct}}(r, s)$
Step 1. $V_r$ receives beacon frame $M$ from $V_s$
Step 2. if $V_s \in T_{\text{direct}}$ then
Step 3. $T_{\text{direct}}(r, s) = \frac{T_s(t) - T_s(t-1)}{100}$
Step 4. else
Step 5. Add $V_s$ to $T_{\text{direct}}$
Step 6.  $T_{\text{direct}}(r, s) = T_{\text{min}}$.
```

- **Indirect trust computation.** Once the direct trust is calculated in Step 1, we add the recommendation trust given by the sender vehicle. The recommendation trust value is the trust value (includes direct and indirect trust) of neighboring vehicles and is evaluated by the recommender forwarder vehicle (V_{rf}). Since the sender can manipulate the receiver by forging false trust values of another vehicle. Therefore, we use cosine similarity to check that the total trust given by the V_{rf} is similar to the receiver V_r. This similarity is considered as a coefficient (denoted by α and bounded between 0 to 1), which can be changed dynamically according to the similarity between the trust value calculated by the sender and receiver vehicle (if α is most similar, then the value tends to 0 while at dissimilarity it tends to 1). After that, if we receive a recommendation for the first time, we add it to the indirect trust list and add it with respect to T_{min}; otherwise, we update it with respect to the previously stored indirect trust.

In first line of Algorithm 2, we evaluate the coefficient value. From Steps 2-4, we update indirect trust for that vehicle that is already in the indirect trust list. From Steps 5-7, we calculate the indirect trust of the newly recommended vehicle.

Algorithm 2. Indirect Trust Computation

```
Input: $T_{\text{tt}}^{r}$, $T_{\text{tt}}^{\text{rf}}$
Output: $T_{\text{indirect}}$
Step 1. $\cos\alpha = \frac{T_{\text{tt}}^{r} \cdot T_r}{|T_{\text{tt}}^{r}| \cdot |T_r|}$
Step 2. for $i = 1$ to $n$ do
Step 3.   if $V_i \in T_{\text{indirect}}$ then
Step 4.     $T_{\text{indirect}}(r, i) = (1 - \alpha) * T_{\text{tt}}^{r}(\text{rf}, i) + \alpha * T_{\text{tt}}(r, i)$
Step 5. else
Step 6.   Add $V_i$ to $T_{\text{indirect}}$
Step 7.   $T_{\text{indirect}}(r, i) = (1 - \alpha) * T_{\text{tt}}^{r}(\text{rf}, i) + \alpha * T_{\text{min}}$.
```

- **Total trust computation.** In the third step, we aggregate direct and indirect trust to calculate the total trust of vehicles. Total trust calculation is different at degree: Cluster Member (CM) and Cluster Head (CH), as shown in Algorithm 3. Total trust evaluates the combined trustworthiness over another vehicle based on direct and indirect trust. Suppose the vehicle is CM; then it calculates total trust only for the direct neighbor vehicle (whose direct behavior is measured). While CH has more responsibility to find a trustworthy vehicle in the environment, it evaluates the total trust of a vehicle whose indirect trust has been calculated but whose direct behavior is unknown.

In the algorithm, Step 1 represents all n vehicles whose total trust is going to be calculated. In Steps 2-3, we take average direct and indirect trust to compute total trust. From Steps 4-6, we evaluate the total trust if indirect trust has been evaluated by vehicle (only applicable to CH). In comparison, direct trust is considered total trust if indirect trust is not available (from Steps 7-9).

Algorithm 3. Total Trust Computation

```
Input:  $T_{\text{direct}}$, $T_{\text{indirect}}$
Output:  $T_{\text{tt}}^{r}$
Step 1.  for $i = 1$ to $n$ do
Step 2.    if $V_i \in T_{\text{direct}}$ and $V_i \in T_{\text{indirect}}$ then
Step 3.    $T_{\text{tt}}^{r}(r,i) = \frac{T_{\text{direct}}(r,\ i) \mp T_{\text{indirect}}(r,\ i)}{2}$
Step 4.     else
Step 5.   if $V_i \notin T_{\text{direct}}$ and $V_i \in T_{\text{indirect}}$ and $V_r \rightarrow \text{DEGREE} == \text{CH}$ then
Step 6.       $T_{\text{tt}}^{r}(r,i) = T_{\text{indirect}}(r,i)$
Step 7.      else
Step 8.    if $V_i \in T_{\text{direct}}$ and $V_i \notin T_{\text{indirect}}$ and $V_r \rightarrow \text{DEGREE} == \text{CH}$ then
Step 9.      $T_{\text{tt}}^{r}(r,i) = T_{\text{direct}}(r,i)$.
```

- **Generate attacker list.** In our proposed model, the cluster head generates an attacker list according to the overall trust calculated using Algorithm 3. First, it computes the average threshold value (by taking the average of the total trust of each neighbor vehicle) and then compares every vehicle's total trust with a threshold. If the neighboring vehicle's trust is below the threshold, then add that vehicle to the malicious list. If the vehicle is previously added to the malicious list and currently, its trust value is above the threshold, then remove that vehicle from the malicious (because sometimes vehicles perform maliciously due to hardware/software failure).

Step 1 checks whether the vehicle is CH or not. From Steps 2-6, we evaluate the average threshold value. At Step 7, we initiate a for-loop to check that all *n* vehicles whose trust value is calculated belong to malicious or not. From Steps 8-10, we add a malicious vehicle, while at Steps 11-12, we remove the vehicle from the malicious list, i.e., currently trustworthy, but in the past, it was trustworthiness.

Algorithm 4 Generate Attacker List

```
Input: $T_{\text{tt}}^{r}$
Output: $L_a(r, i)$
Step 1. if $V_r \rightarrow \text{DEGREE} == \text{CH}$ then
Step 2.  sum=0
Step 3.  for $i = 1$ to $n$ do
Step 4.    $\text{sum} = \text{sum} + T_{\text{tt}}^{r}(r,i)$
Step 5.    count++
Step 6.  $T_{\text{thresh}} = \frac{\text{sum}}{\text{count}}$
Step 7.  for $i = 1$ to $n$ do
Step 8.    if $T_{\text{tt}}^{r}(r,i) < T_{\text{thresh}}$ and $T_{\text{tt}}^{r}(r,i) \neq 0$ then
Step 9.      Add $V_i$ to $L_a$
Step 10.      $L_{\mathrm{a}}(r,i) < T_{\mathrm{tt}}^{r}(r,i)$
Step 11.    else
Step 12.      remove $V_i$ to $L_{\mathrm{a}}$
```

4. Simulation result and analysis

We address the simulation in-depth in this section to test our proposed scheme. First, the simulation environment, followed by the analysis of the results, is defined. Especially its efficiency and performance of how it monitors behaviors, as well as classification of untrusted from trusted ones.

4.1. Simulation environment

We have used NS3 (Network Simulator 3) [27], which is used for network environment simulation, to test the performance of our proposed model. In addition, information related to beacons and emergency alerts has been generated within the network. Every 100 ms, a beacon was transmitted and used to alert the neighbor position while emergency information was produced at a random location within the network. The same emergency alert could be received by a vehicle several times; through this, we improve transmission reliability. The simulation parameter used in the experiment is described in Table 3. We analyze DoSRT performance against the model designed by Hasrouny et al. [25], which is based on the method of evaluating weighted voting trust. For interruptive VANET service experience, many strategies have been introduced to generate Malicious Vehicles (MV). We are concentrating on greedy behavior in this paper, typically based on DoS attacks. The aim is to make vulnerable the operations of the MAC layer and exploitation of accessing the approach of the medium. The goal of malicious vehicles is to reduce the waiting period for quick channel accessing and disallow the service gain to the other honest vehicles [19]. Hence, the channel accessibility restriction and attempts to connect with the medium are affected. The key issue for this attack is that it can be carried out through the authenticated individual (making it more difficult to identify).

Table 3. Simulation parameter

Parameter	Value
Simulation time	30 min
Vehicle distribution	Random
Total number of vehicle	25, 50, 75, 100
Transmission range	60 m
Transmission rate	6 Mbps
Mean speed	30 m/s
Speed deviation	5 m/s
Iterated simulation	25 times

4.2. Result analysis

We have injected 40% of the malicious vehicles in our scenario. To generate a DoS attack, malicious vehicles overload the beacon message to other vehicles. This prevents malicious activity from slowing down or interrupting the service from getting emergency messages. Figs 7 and 8 show the precision and recall of DoSRT, where the adversary transmits a beacon with higher frequency. Table 6 demonstrates that DoSRT's precision and recall reveal that the network achieves high precision and recall when there is a less malicious vehicle. As the number of DoS attackers increased, however, the corresponding precision and recall decreased. This is because the increased number of attackers would result in a high number of beacon generations.

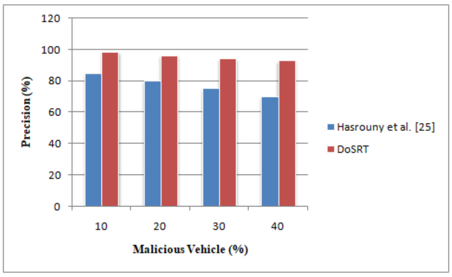

Fig. 7. Precision (%) with respect to percentage of malicious vehicle

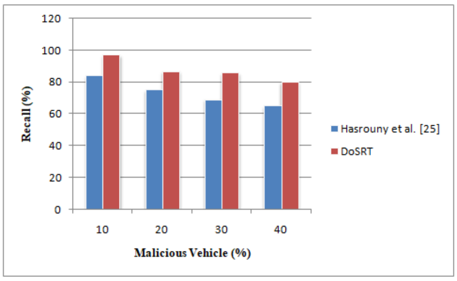

Fig. 8. Recall (%) with respect to percentage of malicious vehicle

This inhibits the ability of the head of the cluster to categorize between trusted and untrusted vehicles. However, it works better in terms of easily distinguishing and classifying trustworthy and malicious vehicles, allowing the vehicle at the initial stage to be revoked the misbehaving vehicle. On the other hand, if we equate DoSRT with Hasrouny et al. [25] trust model, we have found that it is difficult to locate an attacker at the initial stage when a legitimate vehicle is surrounded by a high number of malicious vehicles. Hence, a high number of malicious vehicles degrades the precision and recall value of the Hasrouny et al. [25] trust model. As illustrated, the precision of DoSRT falls from 98.48% to 93.02% if the number of malicious vehicles is from 10% to 40%, while the precision value of the Hasrouny et al. [25] trust model drastically falls from 85% to 70%. As the malicious vehicle increases from 10% to 40%, DoSRT recall decreases from approximately 97.01% to 80%, relative to Hasrouny et al. [25] trust model, where recall falls from 84% to approximately 65%. This demonstrates that DoSRT is effective in dealing with a DoS attack. The F-score of our model, which is one of the critical metrics to test the accuracy of our trust model against the attacker and non-attacker classification, is highlighted in Table 4. The result suggests that, in the presence of 30% malicious vehicles, DoSRT ensures accuracy in terms of an F-score of around 89.71% (Fig. 9), while the Hasrouny et al. [25] trust model achieves an accuracy of approximately 72%.

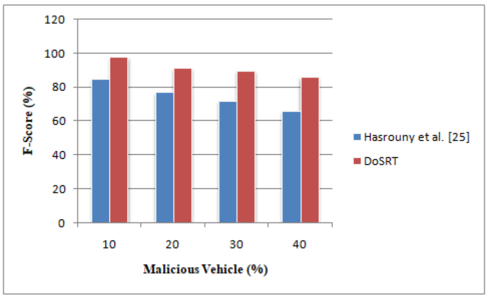

Fig. 9. F-score (%) with respect to percentage of malicious vehicle

Table 4. Comparison between DoSRT and Harsouny et al. [25] w.r.t. metrics (Recall, Precision, F-score, Accuracy, Average trust value)

MV (%)	Recall (%)		Precision (%)		F-score (%)		Accuracy (%)		Average trust value	
	DoSRT	[25]	DoSRT	[25]	DoSRT	[25]	DoSRT	[25]	DoSRT	[25]
10	98.48	85	97.01	84	97.73	85	96	84	9	8
20	96.29	80	86.67	75	91.22	77	86.67	75	8.6	6.9
30	94.11	75	85.71	69	89.71	72	85.33	68	8.1	6
40	93.02	70	80	65	86.02	66	82.66	64	7.9	4.9

Table 4 illustrates the accuracy of our model. Accuracy can reflect how effective the trust model is in detecting a DoS attack. The result suggests that the accuracy of DoSRT falls from 96 % to 82.66 % when malicious vehicles increased from 10 % to 40%, while Hasrouny et al. [25] model decreased from 84 % to 64 % (Fig. 10).

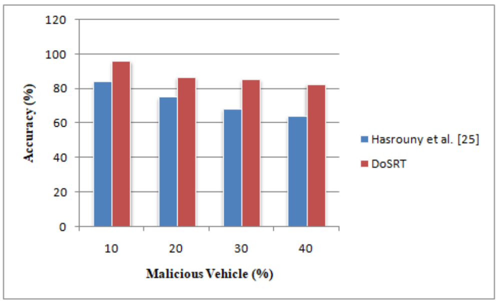

Fig. 10. Accuracy (%) with respect to percentage of malicious vehicle

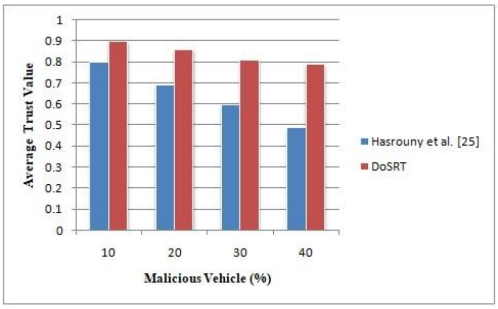

Fig. 11. Average trust value with respect to percentage of malicious vehicle

The efficiency of the DoSRT trust model for classifying vehicles in terms of trust is shown in Table 4. In particular, in the presence of a DoS attack, we measure the behavior of the trust value in Fig. 11. This indicates that when the network is polluted by DoS attackers, the vehicle's trust decreases. This is due to the fact that the higher malicious vehicle restricts the legitimate vehicle from detecting malicious activity when a large number of attackers pollute the network.

However, compared with Hasrouny et al. [25], DoSRT is efficient in identifying and classifying legitimate vehicles in the presence of attackers. This is because DoSRT intelligently distinguishes vehicles conducting malicious behavior, allowing the evaluator vehicle to differentiate between the legitimate and the attack quickly. We can see that when 40% of attackers are present, DoSRT's average trust is 0.79, while Hasrouny et al. [25] assign below 0.5 (if trust is less than 0.5, we assumed the vehicle is untrustworthy).

DoS vs. DDoS attack in VANET. The DoS attack attempts to disrupt the normal functioning of the vehicle by overwhelming it with a flood of illegitimate message traffic. The goal of this attack is to make the targeted system unavailable. The proposed model anticipates a DoS attack by vehicle according to the frequency of messages sent by vehicle. It forms clusters according to the speed deviation factor, with the cluster head generating the attackers' report in order to reduce the overhead of disseminating information. The proposed trust model is effective for DoS attacks carried out by individual attacker vehicles. Another form of DoS attack is known as a Distributed DoS attack (DDoS), wherein a cluster of vehicles launches DoS attacks by gaining simultaneous access to networks. DDoS can be detected by adding the timing of the attack by the attackers together with the frequency of message transmitting by vehicle, which will be considered in an extension of this work.

5. Conclusion

Based on their behavior inside VANETs, we have proposed a trust model for evaluating vehicle trustworthiness. It is a cluster-based approach to trust establishment, where a set of rules identify the malicious or legitimate vehicle and notify others of mitigation against DoS attackers. The efficiency of DoSRT has been checked through extensive simulation, which shows the ability of cluster heads to control cluster members and to recognize between trustworthy and untrustworthy. Finally, we have compared trust evaluations with some existing ones. In our future work, we will extend the trust model by adding features to detect the behavior of distributed denial of service (DDoS) attacks in VANET. Together with this, we will consider scenarios of trust model resistant against Sybil and black hole attack.

References

1. Avasthi, S., T. Sanwal, S. Sharma, S. Roy. Vanets and the Use of IOT: Approaches, Applications, and Challenges. – Revolutionizing Industrial Automation Through the Convergence of Artificial Intelligence and the Internet of Things, 2023, pp. 1-23.

2. Yadav, A. S., A. Singh, A. Vidyarthi, R. K. Barik, D. S. Kushwaha. Performance of Optimized Security Overhead Using Clustering Technique Based on Fuzzy Logic for Mobile ad hoc Network. – Cybernetics and Information Technologies, Vol. **23**, 2023, No 1, pp. 94-109.
3. Raju, M., K. P. Lochanambal. An Insight on Clustering Protocols in Wireless Sensor Networks. – Cybernetics and Information Technologies, Vol. **22**, 2022, No 2, pp. 66-85.
4. Soomro, I. A., H. Hasbullah, et al. User Requirements Model for Vehicular ad hoc Network Applications. – In: Proc. of International Symposium on Information Technology, Vol. **2**, IEEE, 2010, pp. 800-804.
5. Hafeez, M., R. Ahmad, U. Hafeez. The Future of Vehicle Crash Avoidance through Vanets. – Int. J. Adv. Appl. Sci., Vol. **5**, 2018, No 11, pp. 1-15.
6. Sudha, K. S., N. Jeyanthi. A Review on Privacy Requirements and Application Layer Security in Internet of Things (IoT). – Cybernetics and Information Technologies, Vol. **21**, 2021, No 3, pp. 50-72.
7. Komala, C. R., M. Dhanalakshmi, R. Gayathri, R. Aruna, G. R. Thippeswamy. Vanet Backbone in Data Networking and Block-Chain. – Journal of Pharmaceutical Negative Results, 2023, pp. 1539-1553.
8. Wang, X., Z. Ning, L. Wang. Offloading in Internet of Vehicles: A Fog-Enabled Real-Time Traffic Management System. – IEEE Transactions on Industrial Informatics, Vol. **14**, 2018, No 10, pp. 4568-4578.
9. Yadav, A. S., S. Agrawal, D. S. Kushwaha. Distributed Ledger Technology Based Land Transaction System with Trusted Nodes Consensus Mechanism. – Journal of King Saud University-Computer and Information Sciences, 2021.
10. Kumar, A., A. S. Yadav, D. S. Kushwaha. VCHAIN: Efficient Blockchain Based Vehicular Communication Protocol. – In: Proc. of 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2020, pp. 762-768.
11. Hasrouny, H., A. E. Samhat, C. Bassil, A. Laouiti. Trust Model for Secure Group Leader-Based Communications in VANET. – Wireless Networks, Vol. **25**, 2019, No 8, pp. 4639-4661.
12. Zhang, J. A Survey on Trust Management for Vanets. – In: Proc. of IEEE International Conference on Advanced Information Networking and Applications, IEEE, 2011, pp. 105-112.
13. Wu, A., J. Ma, S. Zhang. Rate: A RSU-Aided Scheme for Data-Centric Trust Establishment in Vanets. – In: Proc. of 7th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2011, pp. 1-6.
14. Chen, X., L. Wang. A Cloud-Based Trust Management Framework for Vehicular Social Networks. – IEEE Access, Vol. **5**, 2017, pp. 2967-2980.
15. Wang, X., Z. Ning, X. Hu, E. C.-H. Ngai, L. Wang, B. Hu, R. Y. K. Kwok. A City-Wide Real-Time Traffic Management System: Enabling Crowdsensing in Social Internet of Vehicles. – IEEE Communications Magazine, Vol. **56**, 2018, No 9, pp. 19-25.
16. Ning, Z., X. Hu, Z. Chen, M. C. Zhou, B. Hu, J. Cheng, M. S. Obaidat. A Cooperative Quality-Aware Service Access System for Social Internet of Vehicles. – IEEE Internet of Things Journal, Vol. **5**, 2017, No 4, pp. 2506-2517.
17. Wahab, O. A., H. Otrok, A. Mourad. A Cooperative Watchdog Model Based on Dempster–Shafer for Detecting Misbehaving Vehicles. – Computer Communications, Vol. **41**, 2014, pp. 43-54.
18. Miller, G. S. The Press as a Watchdog for Accounting Fraud. – Journal of Accounting Research, Vol. **44**, 2006, No 5, pp. 1001-1033.
19. El Khatib, A., A. Mourad, H. Otrok, O. A. Wahab, J. Bentahar. A Cooperative Detection Model Based on Artificial Neural Network for Vanet Qos-Olsr Protocol. – In: Proc. of IEEE International Conference on Ubiquitous Wireless Broadband (ICUWB'15), IEEE, 2015, pp. 1-5.
20. Miller, G. S. The Press as a Watchdog for Accounting Fraud. – Journal of Accounting Research, Vol. **44**, 2006, No 5, pp. 1001-1033.

21. Minhas, U. F., J. Zhang, T. Tran, R. Cohen. A Multifaceted Approach to Modeling Agent Trust for Effective Communication in the Application of Mobile ad hoc Vehicular Networks. – IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol. **41**, 2010, No 3, pp. 407-420.

22. Fogue, M., F. J. Martinez, P. Garrido, M. Fiore, C.-F. Chiasserini, C. Casetti, J.-C. Cano, C. T. Calafate, P. Manzoni. Securing Warning Message Dissemination in Vanets Using Cooperative Neighbor Position Verification. – IEEE Transactions on Vehicular Technology, Vol. **64**, 2014, No 6, pp. 2538-2550.

23. Finnson, J., J. Zhang, T. Tran, U. F. Minhas, R. Cohen. A Framework for Modelling Trustworthiness of Users in Mobile Vehicular ad hoc Networks and Its Validation through Simulated Traffic Flow. – In: Proc. of International Conference on User Modeling, Adaptation, and Personalization, Springer, 2012, pp. 76-87.

24. Raya, M., J.-P. Hubaux. Securing Vehicular ad hoc Networks. – Journal of Computer Security, Vol. **15**, 2007, No 1, pp. 39-68.

25. Hasrouny, H., A. E. Samhat, C. Bassil, A. Laouiti. Trust Model for Secure Group Leader-Based Communications in Vanet. – Wireless Networks, Vol. **25**, 2019, No 8, pp. 4639-4661.

26. Rawashdeh, Z. Y., S. M. Mahmud. A Novel Algorithm to Form Stable Clusters in Vehicular ad hoc Networks on Highways. – Eurasip Journal on Wireless Communications and Networking, Vol. **2012**, 2012, No 1, pp. 1-13.

27. Carneiro, Gustavo J. A. M. Ns-3: Network Simulator 3. University of Porto, 2010.

Received: 10.07.2023; Second Version:15.11.2023; Accepted:23.11.2023