Filtering Malicious Messages by Trust-Aware Cognitive Routing in Vehicular Ad Hoc Networks

Ida Mirzadeh, *Member, IEEE*, Mohammad Sayad Haghighi, *Senior Member, IEEE*, and Alireza Jolfaei, *Senior Member, IEEE*

***Abstract*— Vehicular Ad hoc Networks (VANET), as an inseparable part of Intelligent Transportation Systems (ITS), enable data communication between vehicles to promote road safety and traffic efficiency. But adversaries can also spread false information across these networks. Therefore, vehicles' cognitive capabilities with respect to received data must be improved. This requires spatial intelligence and a mechanism to evaluate the trustworthiness of received data. But the dynamic nature of VANETs with intermittent connections, lack of infrastructure and real-time constraints make fulfilling this task very challenging. In this paper, we propose a trust-aware cognitive framework that exploits the redundancies in the exchanged DENM and CAM packets as well as spatial intelligence to filter malicious messages and prevent attackers from disrupting network operation. A novel supplementary mechanism is also presented that applies subjective logic on the pieces of information collected from all network vehicles to detect and isolate malicious entities. To assess the reliability of the proposed scheme, we made extensive comparisons between our model and two others. In the obtained results, our approach outperformed both of them and yielded an accuracy of over 90%, even when 50% of network participants were attackers. The supplementary malicious node detection mechanism of ours similarly yielded high accuracy and F1 scores in the simulations.**

***Index Terms*— Intelligent transportation systems, vehicular ad hoc networks, cognitive networking, trust management, intrusion detection, cyber security.**

I. Introduction

THE number of accidents has grown along with the increase in the number of vehicles. Each year, out of one million, 174 people lose their lives in road accidents [1]. Moreover, between 20 to 50 million people have become disabled due to injuries caused by accidents. According to studies, if appropriate warnings are given to drivers beforehand, 60% of accidents can be avoided [2], [3]. Vehicular ad hoc networks are a subset of mobile ad hoc networks, wherein each vehicle acts as a node. Each node can interact with other moving nodes or with Road Side Units (RSU) that are placed along roads.

Manuscript received 25 July 2021; revised 16 January 2022; accepted 28 February 2022. This work was supported in part by IPM under Grant CS1401-4-771. The Associate Editor for this article was Y. Zhang. *(Corresponding author: Mohammad Sayad Haghighi.)*

Ida Mirzadeh is with the School of Electrical and Computer Engineering, University of Tehran, Tehran 1439957131, Iran (e-mail: a.mirzadeh@ut.ac.ir).

Mohammad Sayad Haghighi is with the School of Electrical and Computer Engineering, University of Tehran, Tehran 1439957131, Iran, and also with the School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran 19395-5746, Iran (e-mail: sayad@ut.ac.ir).

Alireza Jolfaei is with the College of Science and Engineering, Flinders University, Adelaide, SA 5042, Australia (e-mail: alireza.jolfaei@flinders.edu.au).

Digital Object Identifier 10.1109/TITS.2022.3191634

This type of network has security vulnerabilities, just like any other networks. However, due to its mobile, dynamic, distributed nature and other challenges which are thoroughly discussed in [4], [5] many of the approaches and countermeasures that are applicable to other networks, such as MANETs, will not work in VANETs. The majority of solutions developed to improve security of such networks are based on cryptography. However, such cryptographic methods are not usually resilient to insider attackers. Compromised insiders can disseminate false information in the form of data or safety packets [6]. For example, a car can falsely report an immediate brake or road block, which in turn can cause accidents for other vehicles. But the chance of repeated interactions with the same peers is low in VANETs. Due to mobility, the fact that a node has communicated with another node provides no guarantee that such a communication happens again the future [7]. Many of the solutions presented for VANETs assumed nodes have enough prior knowledge about each other from their past interactions, which is an invalid assumption.

In this paper, we aim to give vehicles cognitive abilities so that during routing, they can distinguish between fake and legitimate messages on their own. More precisely, we propose a cognitive trust management scheme for VANETs that can help vehicles evaluate the trustworthiness of received packets. Any scheme compatible with VANETs must be 1) Decentralized, since access to infrastructure is limited. Each node must have the cognitive abilities to evaluate the trustworthiness of received data autonomously; 2) Scalable, so that it can perform well under sparse and dense network conditions; and 3) Lightweight, and not bring extra overhead to the network, as each node must make decisions in a real time manner.

Our scheme exploits the existing data redundancy in the architecture and messaging pattern of VANETs to achieve the above aims. Redundancy in data is helpful in the promotion of trust, because if information is received from multiple paths (and its pieces are inconsistent due to existence of both benign and malicious nodes in the network), the possibility of making the right decision can be increased [8], [9]. Based on this, the information existing in Cooperative Awareness Messages (CAM) and Decentralized Environmental Notification Messages (DENM) is used to give nodes environmental cognitive abilities. Furthermore, we propose a method that employs collective intelligence to identify malicious nodes in network. This can be used by an additional unit that monitors the whole network for smooth operation and takes compensating actions like revoking the credentials of dishonest nodes.

The simulation results show that with the help of our cognitive trust management scheme, nodes are able to verify the trustworthiness of received packets accurately. In addition, malicious vehicles are detected with an accuracy of over 90%. The contributions of this paper are summarized as follows:

- Based on subjective logic, a VANET message filtering mechanism is proposed that reduces malicious nodes effect by taking network data redundancies into account. The resultant model assesses the trust to original packets as well as relayed ones by using both event and beacon messages. For relayed packets, a consistency verification module is proposed that calculates trust based on data freshness, reliability and routing path length. For original packets, reporting nodes' behavior is taken into account.
- As an additional effort, a malicious node detection module (MND) is proposed with which misbehavior is detected in a centralized offline manner in order to stabilize network by node revocation.

The rest of the paper is organized as follows: In Section II, the literature around trust management in intelligent transportation systems is reviewed. Section III gives the assumptions and the proposed cognitive trust management built upon them. Section IV evaluates our model with extensive simulations. Finally, the paper is concluded in Section V.

II. Related Work

Trust management schemes can be categorized into three groups: entity-based, data-oriented and hybrid approaches. In entity-based approaches the focus is on measuring the trust to an arbitrary node, whereas data-oriented mechanisms focus on the credibility of the piece of data itself. When this approach uses nodes reputation as a parameter in data trust evaluation, it is considered as a hybrid scheme.

Reference [10] proposed a data-oriented trust model that calculates a trust value for each packet based on security status of the vehicle, type of the vehicle, the reported event and other dynamic factors. Security status functions indicate whether the vehicle is authorized or unauthorized. This model assumes that each vehicle has a type, thus has a default trustworthiness which affects the final trust value. After a packet trust value is calculated, it will be aggregated with others using majority voting, weighted voting, Bayesian or Dempster-Shafer theory.

The authors of [11] suggested a data-oriented approach that showed better performance in comparison with Bayesian inference. In their method, first, a two-step clustering is done on packets to group the relevant packets. Then, based on content similarity and routing path similarity, the trust value of each group is calculated. The cluster that has the maximum trust value is selected. Path similarity is a kind of penalty for the messages belonging to the same cluster. As the similarity of paths increase, the support that they give each other decreases.

In [12], the trust to each packet is calculated based on four parameters. A message is validated in terms of time and location. If the provided time and location information is deemed false or invalid, its parameters will be zeroed. The trust to a message depends on the proximity of time and the proximity of space between the receiver and the sender.

The authors of [13] presented an approach called "ERS," which is composed of two parts. The first component calculates the reputation of accident (report) and the second component calculates the confidence level of accident. The reputation of an accident depends on the sensors of the vehicle that detected the accident. The reliability of the report also depends on the number of machines that detected the accident. The final decision is made based on the two thresholds corresponding to these two parameters.

Trust value in [14] is defined based on node role. A node can be event reporter, event observer or event participant. For reporters, trust to the packet depends on the frequency of detected event(s). Observers are one hop neighbors of reporters and can observe their behavior. Each reported trust value has a weight that is function of node"s behavioral deviation. Participating nodes calculate trust value in a similar way. At the end, if the final trust value is more than a predefined threshold, the vehicle will send the message to others.

An RSU-aided mechanism was introduced in [15]. When an event is detected, vehicles calculate observation and confidence values of the event. Observation factor is dependent on the number of sensors that reported the incident, their distances to the incident and a weight that pertains to the vehicle's default trustworthiness. In addition, for every evidence a feedback factor is calculated which decreases by time. But if a supporting evidence is found, this value will increase. These two factors will be used to calculate the final trust.

Researchers in [16], [17] and [18] employed an entity-based trust model in their work. In the framework presented in [16], after an accident happens, when a vehicle receives a number of reports related to the accident, it selects advisors for guidance. The advisors vary depending on the sensitivity of the incident as well as time constraints. They will be sorted based on their role and communication history, then a request for advice is sent to a few nodes. After aggregating the responses, if a majority consensus is reached, the most trusted report will be accepted, otherwise the report of the node with the highest role and experience will be accepted.

TRIP is the method presented in [17]. In this method a reputation value is calculated by using direct experiences, opinion of other nodes, and if available, opinion of Road Side Units (RSU). Based on the calculated value, a vehicle is labelled by TRUST, NOT TRUST or $+/-$ TRUST. When it has been decided which reports contain true information, reputation of recommenders is updated. In addition, each event has its own intensity level. For example, critical messages can only be accepted from vehicles whose labels are TRUST.

Huang *et al.* [18] used a number of local authorities (LA) with high processing capability at the edge. Vehicles can directly communicate to them and query a node"s reputation. These LAs can be connected to the global database via the Internet. Each vehicle sends its opinion about other neighboring vehicles to the local authority. LA aggregates all the reputation segments with their corresponding weights to update the reputations. The weight of a reputation segment is calculated based on its familiarity, similarity and timeliness. Subjective logic is used to model and aggregate the opinions.

The goal of writers in [19] was to detect ON-OFF attacks. To do so, they updated reputation periodically rather than after each interaction. In this approach, there are direct and indirect trust values that constitute the total trust. As the number of interactions between two vehicles increases, the weight of the direct part gets higher. For indirect trust, opinions of the node about its neighbors and their direct trust values are used. Deviation between two consecutive trust values is used to detect an ON-OFF attack. Similar efforts in the Internet of Thing context was done in [20], [21].

Reference [22] suggested a hybrid trust model for VANETs. In this system, each node attaches its opinion (trust, not trust) along with the confidence value to the packet. For decision making, trust values with their corresponding confidence values will be aggregated. It is then decided if the node should accept/relay the packet. Each peer's' trust is evaluated, either role-based or experience-based. The roles and the trust values are fixed by an offline authority. If a peer does not have a role, experience-based trust is used. This quantity reflects the trustworthiness of a peer and gets updated regularly.

Beacon messages are used in [23] to assess the trust value. This value is dependent on the opinions of RSUs and a combined trust indicators which are calculated based on direct and indirect messages. Aggregation is done using the Dempster-Shafer Theory (DST) operator. This theory is similarly used in [24], [25]. Reference [24] uses collaborative filtering to estimate trust. Each node has a vector of trust ratings. When a node wants to estimate trustworthiness of an unknown node, it uses opinions of the N most similar nodes who know that node. Cosine similarity determines the closeness of the rating vectors of the node and its neighbors. The final trust is calculated based on the similarity rate to the recommender node, trust values of recommender and recommended nodes and the actual trust value to the unknown node.

Reference [25], based on previous beacon messages, obtains a prediction for the target vehicle location. Then, cosine similarity is calculated between the predicted vector and the newly received beacon. This computation can be done for multiple beacons and the final beacon trust value is calculated using time-based weighting method. Trust to an event is composed of direct and indirect trust values. Final trust value equals to the direct trust value if the packets are received directly, otherwise it depends on the opinions of others. Vehicle's reputation value, beacon trust value and event trust value are mixed together to form a composite trust to the received message.

III. The Proposed Trust-Aware Cognitive Routing

Upon the occurrence of an event, sensors of the vehicles driving close by will detect the event and send the corresponding DENM data messages. Neighboring vehicles will receive the message and rebroadcast it to their neighbors. Therefore, all the messages reporting the incident will be propagated into the network by Geobroadcast protocol and each vehicle will receive multiple versions of the message. Moreover, based on ETSI TC ITS standard, a CAM packet shall be sent periodically by each vehicle to its one-hop neighbors. This message contains situational information about the transmitting vehicle. Exchange of CAM and DENM messages takes place in the lower layers of VANET architecture. In addition to Vehicle to Vehicle (V-V) messaging, communication can be done between RSUs and vehicles. RSUs can be connected to each other and share information with a Central Server (see Fig. 1). The vehicle that receives this information must assess its trustworthiness to take appropriate actions. In this section, we propose a cognitive trust management scheme that can be used by vehicles to evaluate the trustworthiness of exchanged messages during routing. Our framework consists of two main pieces, a consistency verification component and a behavioral verification component, that are parts of our Cognitive Data Trust Management (DTM) module (see Fig. 2). The former is used to analyze the amount of conflict in the received data; whereas the latter is used to check the credibility of the main reporters that sent the alarm messages. We also introduce a method to detect malicious vehicles which is explained in Section III-D. It keeps the entire network stable.

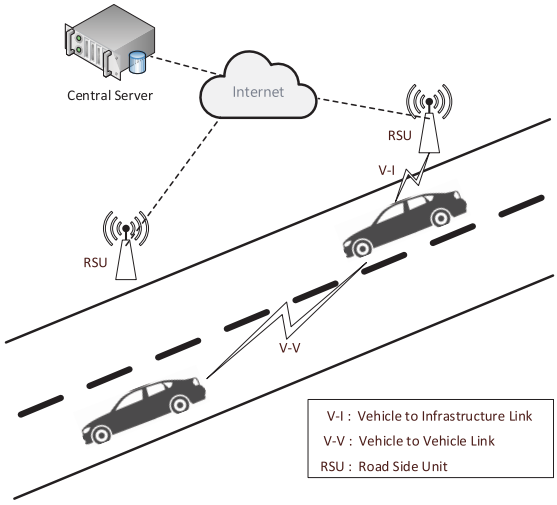


Fig. 1. A VANET architecture with V-V and V-I communication links.

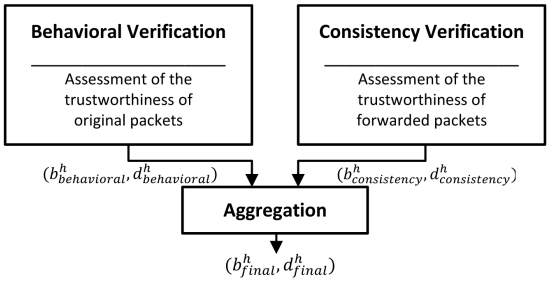


Fig. 2. Components of our cognitive data trust management (DTM) model.

A. V2V Messaging

CAM and DENM are the two main messages handled by the facility layer of the ETSI protocol stack. CAM are distributed within VANET network (802.11p) to inform other vehicles about the situational updates of vehicle [3]. The main purpose of DENM messaging is to announce any abnormal event.

Each vehicle's situational data takes about 27 bytes, whereas a beacon size is around 100 bytes (see Fig. 3). It has been advised that a packet of 200 bytes is appropriate, but larger packet sizes can be used for some applications [26].

Algorithm 1 CAM Generation Algorithm

for every T seconds **do** ▷ $T = 1/f$, f= frequency of cam sending between 0.1s to 1s
  if $|heading - prv_{cam}.heading| \geq thr_{heading}$ **or** $|position - prv_{cam}.position| \geq thr_{position}$ **or** $|speed - prv_{cam}.speed| \geq thr_{speed}$ **then**
    $CAM_{bag} \leftarrow new\ CAMBag(id)$
    $CAM_{Msg} \leftarrow new\ CAMMsg(speed, head, pos, time)$
    Add CAM_{Msg} to CAM_{bag}
    $prv_{cam} \leftarrow CAM_{Msg}$
    Send CAM_{bag}
  end if
end for

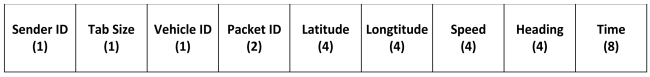


Fig. 3. The payload format of a vehicle's beacon.

As [27] states, additional information can be added to beacon messages. Thus, we insert extra data into CAM messages by piggybacking each vehicle's SIF (vehicle's situational information as depicted in Fig. 3). In piggybacking, upon receiving a packet, the SIF of the current vehicle is attached to it [28]. Thus, it contains SIFs of the vehicles that the packet has traversed through. After attachment, packet will be relayed until its Time to Live (TTL) expires. In our approach, CAM messages propagate in a multi-hop manner, similar to DENM messages (with lower TTL), and at every node, a 27 bytes of SIF will be piggybacked to the message. The same approach is adopted for DENM messages. This will add the data redundancy we need but at the cost of higher overhead.

Details of the messaging is described in algorithms 1,2,3, and 4. Periodically, when situational status of a vehicle changes beyond a certain threshold, a CAM message is sent (Alg. 1). Besides, after receiving other vehicles' beacon messages, each vehicle must append its own SIF to the received CAM packet and broadcast it to its neighbors (Alg. 2). Upon detecting an event, a DENM packet containing event's detail alongside the vehicle's SIF is sent. This DENM packet is cached and is forwarded if a new vehicle comes into the transmission range of the vehicle (Alg. 3). In Alg. 4, upon receiving a DENM packet, SIF of the current vehicle will be appended to the packet if its TTL has not expired. This received packet is also cached and will be re-sent when new vehicles come around.

B. Consistency Verification

The main source of information for this part is DENM messages. When a malicious node has some control over network traffic, the amount of corrupt information will increase and the accuracy of trust evaluations will decrease. We adopt a two-stage filtering mechanism to neutralize the side effects.

1) Node-Disjoint Path Filtering: When a data packet traverses through more distinct nodes, the probability that the receiver can obtain the original data is higher. Similarly, when more packets are received through the same set of nodes, no additional information can be obtained. Therefore, we use the concept of node-disjoint paths in graph theory to filter messages first. Node-disjoint paths can be extracted in a graph by Edmonds–Karp algorithm in $O(VE^2)$. But in our approach, since every packet carries its traversed path, a simple checking for common relays can give us node-disjoint paths in $O(V^2)$.

Algorithm 2 CAM Forwarding Algorithm

while true **do**
  if $n_{hops}(msg) < cam\_TTL$ **and** $!isExist(id, msg)$ **then**
    $CAM_{bag} \leftarrow getCAMBag(msg)$
    $CAM_{Msg} \leftarrow new\ CAMMsg(speed, head, pos, time)$
    Add CAM_{Msg} to CAM_{bag}
    $prv_{cam} \leftarrow CAM_{Msg}$
    Resend CAM_{bag}
  end if
end while

Algorithm 3 DENM Generation Algorithm

while True **do**
  $Strength \leftarrow getStateSensor(id)$
  if $Strength > 0$ **then**
    $RoadId \leftarrow getRoadPosition().getRoadId(id)$
    $DENM_{Msg} \leftarrow new\ DENM(time, RoadId, Strength)$
    $DENM_{SI} \leftarrow new\ SI(id, speed, head, pos, time)$
    Add $DENM_{SI}$ to the bag of $DENM_{Msg}$
    $Cached_{orig\_msg} \leftarrow DENM_{Msg}$
    Send $DENM_{Msg}$
  end if
  if a new node is in range **then**
    if $Cached_{orig\_msg} \neq null$ and $!resend$ **then**
      Re-send $Cached_{orig\_msg}$
      $resend \leftarrow True$
    end if
    if $Cached_{msg} \neq null$ **then**
      Re-send $Cached_{msg}$
    end if
  end if
end while

Algorithm 4 DENM Forwarding Algorithm

while True **do**
  if $n_{hops}(msg) < denm\_TTL$ **and** $!isExist(id, msg)$ **then**
    $DENM_{bag} \leftarrow getDENMBag(msg)$
    $DENM_{SI} \leftarrow new\ SI(id, speed, head, pos, time)$
    Add $DENM_{SI}$ to $DENM_{bag}$
    $Cached_{Msg} \leftarrow Msg$
    Resend Msg
  end if
end while

2) Second Phase Filtering: We filter the packets with paths that are sub-paths of other packets. Due to broadcast nature of routing in VANETs, this may happen, as in Fig. 4. Without this step, a node like DST in the top figure will receive multiple corrupt packets, and merely two uncorrupted packets, and this leads to an incorrect decision. This type of packets with longer paths, despite the possibility of coming from node-disjoint paths, must be removed before decision making.

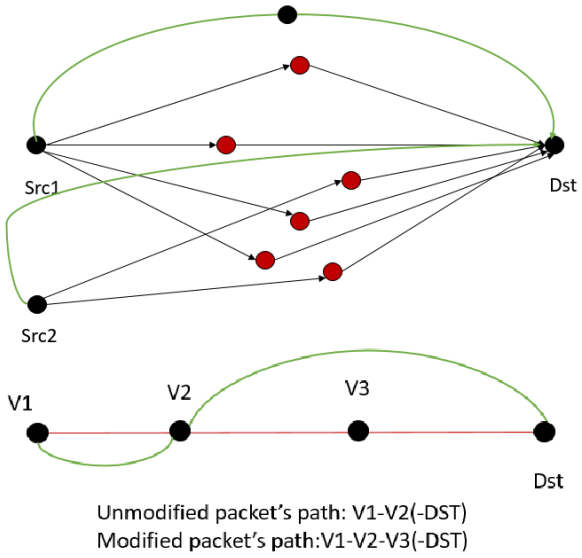


Fig. 4. Two sample graphs showing the travelling paths of VANET packets in multiple-source and single-source scenarios.

3) Trust Modeling: After filtering the packets, to verify the hypotheses h and $\bar{h}$, we need to model our opinion regarding the trustworthiness of the reported event. Hypothesis h means the reported incident has actually happened, whereas $\bar{h}$ means it has not. The beliefs about h and $\bar{h}$ are calculated as follows:

$$Belief_E^h = \frac{1}{c} \sum_{q \in \Pi_h} \psi(q).\chi(path_q) \tag{1}$$

$$Belief_E^{\bar{h}} = \frac{1}{c} \sum_{q \in \Pi_{\bar{h}}} \psi(q).\chi(path_q) \tag{2}$$

where the subscript E stands for the event message or DENM, Π_h is the set of packets consistent with h and $\Pi_{\bar{h}}$ is the set of packets consistent with $\bar{h}$. ψ and χ are functions that measure path value and packet value, respectively. c is the number of all the paths that packets congruent to hypothesis h or $\bar{h}$ travel through (the packets which have been received by the vehicle (Π) or they must have been received but were lost due to malicious acts of attackers or network status (Φ). This set of paths/packets expected to be received by a vehicle are derived based on CAM messages).

a) Path value χ: Function χ is dependent on two parameters: path reliability and path length. Path length is an important parameter, as trust value decays over multiple hops. If x is the percentage of malicious nodes in the network, the probability of data modification/manipulation over l hops is,

$$p_{mod} = 1 - (1 - x)^l \tag{3}$$

which gets higher as the path length increases. The deciding node receives multiple DENM and CAM messages. Based on them, it must deduce whether the event has actually happened or not. Conflicting packets are also received, which increase the disbelief to h. But this situation occurs easily in a dynamic network like VANET. For example, nodes may get stuck behind a red light that leads to connection drops. Path reliability is added to the formula to lower wrong decisions when connections are intermittent. Function χ is defined as,

$$\chi(path_q) = R_{path_q} \times e^{-\alpha_l * \frac{l - l_{min}}{l}} \tag{4}$$

$$R_{path_q} = \min_{\forall link \in path_q} (s_{link}) \tag{5}$$

$$s_{link} = \max(0, 1 - \frac{d}{r}) = \max(0, 1 - \frac{loc_{v_i}^{t_e} - loc_{v_j}^{t_e}}{r}) \tag{6}$$

and $l = |path_q|$ is the path length. With Eq. (4), link stability can be calculated. Parameters α_l and l_{min} are the normalization factor and the minimum path length of the received packets, respectively. r is the transmission range and d is the distance between vehicles at the time of the event (t_e), that is the distance between vehicle v_i's location (denoted as loc_{v_i}) and vehicle v_j's (location denoted as loc_{v_j}).

b) Packet value, ψ: Our framework, like other research efforts [10], [12], captures the dynamicity of environment with location/time closeness. Receiver time closeness metric assesses the freshness of received data in the time dimension.

$$tc_{rc} = \begin{cases} 0, & \text{if } t_{received} - t_{event} \geq \delta_t' \\ 1 - \frac{t_{received} - t_{event}}{\delta_t'} & \text{otherwise} \end{cases} \tag{7}$$

Similarly, receiver location closeness metric assesses the freshness of received data in the space dimension.

$$lc_{rc} = \begin{cases} 0, & \text{if } |loc_{received} - loc_{event}| \geq \delta_d' \\ 1 - \frac{|loc_{received} - loc_{event}|}{\delta_d'} & \text{otherwise} \end{cases} \tag{8}$$

Reporter time closeness metric assesses the freshness of the report created about the event. The report is presumably created after the event has happened.

$$tc_{rp} = e^{-\frac{t_{report} - t_{event}}{\delta_t}} \tag{9}$$

With δ_t', δ_d' and δ_t, one can adjust the sensitivity of these metrics to time and location closeness. Having the above in mind, $\psi(q)$ can be written as,

$$\psi(q) = \beta_1 tc_{rc} + \beta_2 lc_{rc} + \beta_3 tc_{rp} \tag{10}$$

where $\beta_1 + \beta_2 + \beta_3 = 1$. Now, disbelief to h and $\bar{h}$ will be,

$$Disbelief_B^h = \frac{\lambda_B}{c} \sum_{\phi_i \in \Phi_h} \chi(\phi_i) \tag{11}$$

$$Disbelief_B^{\bar{h}} = \frac{\lambda_B}{c} \sum_{\phi_i \in \Phi_{\bar{h}}} \chi(\phi_i) \tag{12}$$

Subscript B stands for the beacon message or CAM, and λ_B is a normalization parameter to adjust the opinion derived from beacon messages (CAM) to event messages (DENM) numerically. Although DENM messages should be the basis of judgement here, sometimes there is not enough evidence to do so. Therefore, we take beacon messages into account too. Φ_h stands for the paths/routes from which we received beacons and based on the graph, expected to receive data packets supporting h. Similarly, $\Phi_{\bar{h}}$ stands for the paths constructed

TABLE I
EVENT - BEHAVIOR PAIR IN VEHICULAR ENVIRONMENTS

Event	Expected Behavior
EEBL	Sudden Slow Down
PCN	Speed Decrease & Lane Change
RHCN	Speed Decrease & Route Change
RFN	Speed Decrease
SVA	Lane Change & Speed Decrease
CCW	Slow Down
CRN	Lane Change
CL	Lane Change & Slow Down

based on beacon signals through which we expected to see evidences supporting $\bar{h}$. In the above, c is calculated as,

$$c = |\Phi_{\bar{h}}| + |\Phi_h| + |\Pi_{\bar{h}}| + |\Pi_h| \tag{13}$$

Finally, belief and disbelief to messages consistency will be,

$$Belief^h_{consistency} = b^h + d^{\bar{h}} = b^h_E + d^{\bar{h}}_B \tag{14}$$

$$Disbelief^h_{consistency} = b^{\bar{h}} + d^h = b^{\bar{h}}_E + d^h_B \tag{15}$$

b is the notion belief in Subjective logic and d is the notion of disbelief. Index B refers to the disbelief found via CAM/beacon messages and E refers to the belief obtained through DENM ones.

C. Behavioral Verification

All the formulation discussed in the previous section is valid if the original reporters (i.e. the vehicles that claim to have directly sensed an abnormal condition), behave honestly. If not, no matter how honest relays are, receiving nodes will get corrupted data. To assess the reliability of original reporters, we make a verification module that models the credibility of reporters in the form of an evidence tuple. When an accident happens, if a node is close enough, its behavior must change accordingly. Reference [29] presents actions corresponding to different types of events/alarms in VANETs (see Table I [29]).

Each node must assess the credibility of original senders upon receiving their messages. Credibility is modeled as $W = (b, d, u)$, in which belief is the ratio of positive evidences to all of the evidences, and disbelief is the ratio of negative evidences to all of the evidences. Nodes receiving a fake DENM message will compute the following relations based on the CAMs received within a time window of 30 seconds.

$$b^h_{bevioural} = \frac{p}{p+n+2} \tag{16}$$

$$d^h_{bevioural} = \frac{n}{p+n+2} \tag{17}$$

Positive evidences (p) can be derived from CAM multihop messages, and negative evidences (n) are like unusual distance between reporter and its one-hop neighbor. Finally the opinion from consistency component will be combined with the behavioral verification component. For this purpose, we used transitivity operator of subjective logic [30].

$$b^h_{final} = b^h_{consistency}.b^h_{behavioral} \tag{18}$$

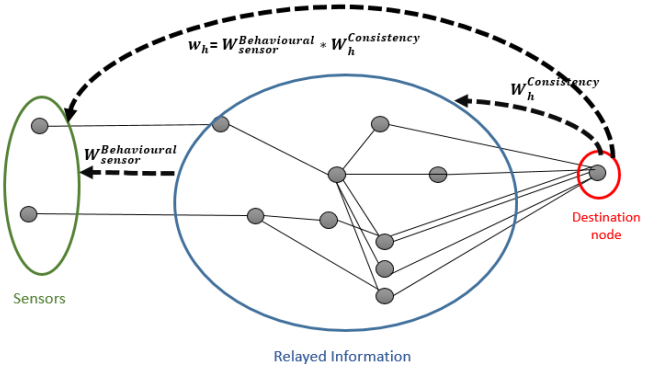


Fig. 5. Aggregation of consistency and behavioral opinions on data.

$$d^h_{final} = d^h_{consistency}.b^h_{behavioral} \tag{19}$$

$$u^h_{final} = d^h_{behavioral} + u^h_{behavioral} + u^h_{consistency}.b^h_{behavioral} \tag{20}$$

Fig. 5 shows this process. Based on the final opinion, each node autonomously decides whether or not to accept h.

If a capable malicious node acts as a sensor and reports a fake incident, it will not just fake the DENM message, since it can be easily detected. In this part, we try to act as an attacker to see what kind of information may get corrupted following an attack. An attacker knows that its neighbors expect it to slow down (or any other appropriate behavior) upon reporting an accident. Hence, it decreases its speed accordingly. Based on deceleration, it has ΔT seconds to get to zero speed, as expected by others. ΔT is calculated as v_{last}/a_{dec}, where v_{last} is the reported speed in the latest beacon prior to the fake DENM message. a_{dec} is the minimum deceleration of a vehicle. The attacker takes a fake $S \leq \Delta T$ to decrease its speed to zero, but the relation between speed, travelled distance (Δk) and elapsed time must be maintained. Next, the attacker forges its position based on the last reported position using the Haversine equations [31].

$$lon_{fake} = lon_{last} + (\frac{180}{\pi})(\frac{(\frac{d_x}{R})}{\cos(lat_{last})}) \tag{21}$$

$$lat_{fake} = lat_{last} + (\frac{180}{\pi})(\frac{d_y}{R}) \tag{22}$$

$$\Delta k = distance(pos_{fake}, pos_{last}) \tag{23}$$

R is the earth radius, θ is the heading of the vehicle. d_x is $\cos(\theta).\Delta k$ and d_y is $\sin(\theta).\Delta k$. The attacker can forge DENM packets and spoof its location in the next CAM packets. But it cannot forge other vehicle's SIFs. This leads to cases in which the distance between the attacker and its neighbors does not match the transmission range or the minimum space between vehicles, thus we find negative evidences.

D. Malicious Node Detection (MND)

In order to make network nodes behave honestly, detection of malicious entities is necessary. The knowledge of a single node is limited. It is more practical to delegate this task to a module that has access to more information (e.g. central server). This task can be done in an offline manner. For every event, multiple packets are distributed. Each data sample is

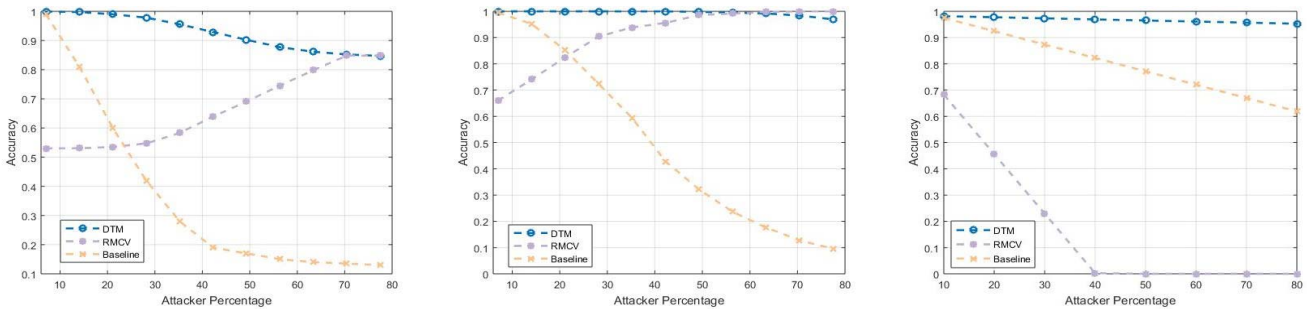


Fig. 6. Effect of malicious node ratio on (a) DTM–alteration pattern, (b) DTM–bogus info, and (c) DTM–suppression pattern.

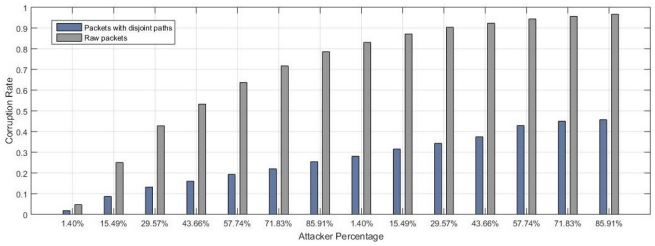


Fig. 7. Corrupt packet ratio versus network attackers percentage.

considered a piece of information. But this piece may convey false data. All of the paths/packets that pertain to a single event are considered a group evidence for the MND task. We show this evidence by e, and its path sub-evidences by I. After a while, we collect a set of such evidences. A packet which is traversed through network up to a destination is relayed by different nodes. If one of them acts mischievously, the entire data will be corrupt and I is labelled as "Corrupted" or "0".

$$e_i = \{I_1, I_2, \ldots, I_m\}$$
$$I_j = \{Q_j = (s_1, s_2, s_3, \ldots, s_d), L_j \in \{0, 1\}\}$$

Let us say we have m pieces of information per evidence. Each information has a sequence of nodes that forms its propagation path, and each piece of I_j can be corrupted or trustworthy which is shown by label L_j. The opinion (b_h^A, d_h^A) about the honesty/trustworthiness of node A is calculated by the following formulas. Note that [.] stands for Iverson bracket.

$$b_h^A = \frac{\sum_{i=1\&L_i=1}^{m} \sum_{j=1}^{d-1} [s_{ij} = A]}{\sum \sum_{j=1}^{d-1} [s_{ij} = A] + 2} \tag{24}$$

$$d_h^A = \frac{\sum_{i=1\&L_i=0}^{m} \sum_{j=1}^{d-1} [s_{ij} = A]}{\sum \sum_{j=1}^{d-1} [s_{ij} = A] + 2} \tag{25}$$

The value of belief about A's honesty is the ratio of its participation in forwarding trustworthy packets, i.e. $p/(p+n+2)$, and the value of disbelief is the ratio of its participation in the paths with corrupt packets. However, there is a problem with Eq. (24) and (25). When a node is far from the original sender, its control over network traffic manipulation gets lower, and also the chance that it receives data which is modified by previous nodes increases. Many negative evidences originating from further up along the path can frame an honest node that relays data down the path without modification. If we use unweighted evidences to classify this node, it will probably be falsely labelled as a malicious. But instead of simply collecting positive/negative evidences, we can adopt a weighted approach. In the weighted approach, we consider node's control over relaying the packet by a weight, i.e. $(d+1-j)/d$. Using this approach can mitigate the abovementioned problem, since we add less value to evidence by increasing uncertainty. The weight is 1 for the source node and 0 for the destination node.

$$b_h^A = \frac{\sum_{i=1\&L_i=1}^{m} \sum_{j=1}^{d-1} [s_{ij} = A] * \frac{d+1-j}{d}}{\sum \sum_{j=1}^{d-1} ([s_{ij} = A] * \frac{d+1-j}{d}) + 2} \tag{26}$$

$$d_h^A = \frac{\sum_{i=1\&L_i=0}^{m} \sum_{j=1}^{d-1} [s_{ij} = A] * \frac{d+1-j}{d}}{\sum \sum_{j=1}^{d-1} ([s_{ij} = A] * \frac{d+1-j}{d}) + 2} \tag{27}$$

Eq. (26), and Eq. (27) are calculated per evidence. Then, the evidences are aggregated using the subjective logic consensus operator, which is associative and accumulative [32].

$$b^{e_1 \Diamond e_2} = \frac{b^{e_2}.u^{e_1} + b^{e_1}.u^{e_2}}{u^{e_1} + u^{e_2} - u^{e_1}.u^{e_2}} \tag{28}$$

$$d^{e_1 \Diamond e_2} = \frac{d^{e_2}.u^{e_1} + d^{e_1}.u^{e_2}}{u^{e_1} + u^{e_2} - u^{e_1}.u^{e_2}} \tag{29}$$

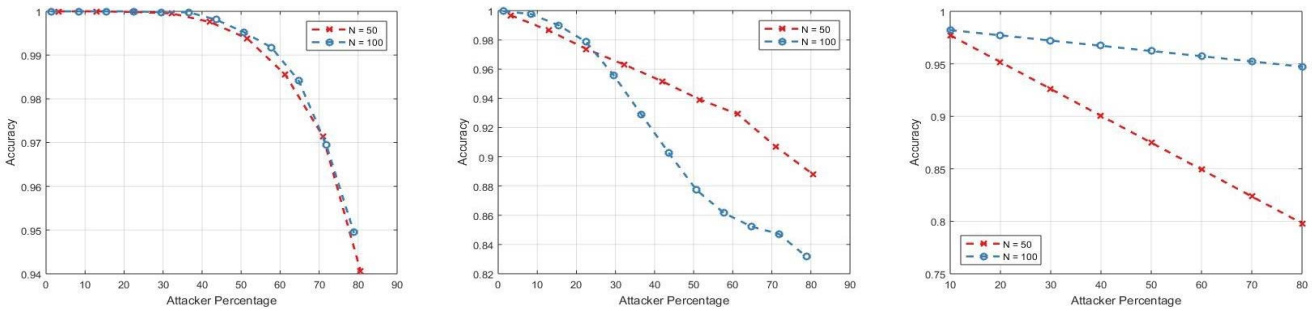
and finally after aggregating all the evidences,

$$E_h^A = b_h^A + u_h^A.a_h \tag{30}$$

E_h^A is the value that is used for decision making. It will be compared against a threshold. If E_h^A is above the threshold, node A is labelled as honest, otherwise, malicious. The threshold we use in our framework is the harmonic mean of all of the expected values of trustworthiness (E_h^* of all the nodes),

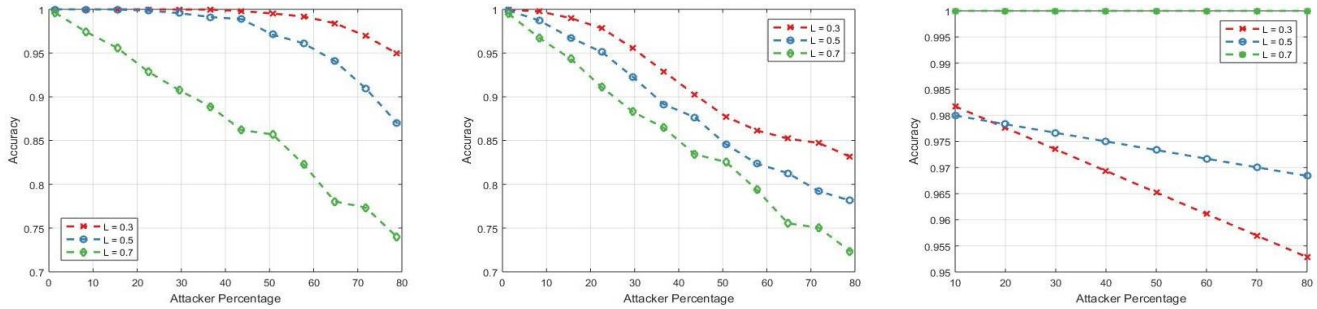
$$\eta = \frac{n}{\frac{1}{E_h^1} + \frac{1}{E_h^2} + \cdots + \frac{1}{E_h^n}} \tag{31}$$

A static threshold is not appropriate, because when the number of malicious nodes increases, the overall trust value to an arbitrary host node decreases, since it will be a part of their paths in fake message delivery. As the final trust values, which are between 0 to 1, get smaller, the boundary between the trust values of an honest node and a malicious one becomes narrower. Harmonic mean uses inverses of trust scores. It is a better discriminator in such cases and fits our purpose.



(a) (b) (c)

Fig. 8. Effect of node density on the accuracy of the proposed model under (a) suppression attack, (b) alteration attack, and (c) bogus information.



(a) (b) (c)

Fig. 9. Effect of loss rate on accuracy under (a) suppression attack, (b) alteration attack, and (c) bogus information attack.

IV. Performance Evaluation

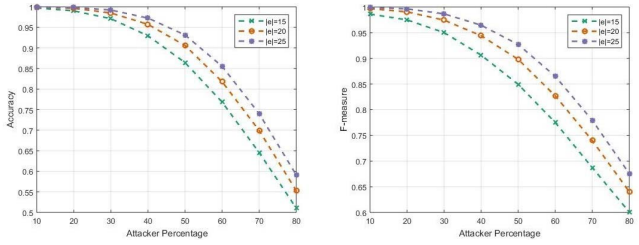
A. Simulation Tools & Settings

In this section, we evaluate the performance of our trust management scheme. To simulate VANET networks, we use VSimRTI [33]. Different simulators are coupled in this framework such as SUMO, ns-3, and OMNET++. We take SUMO [34] as the mobility simulator. We use the map of Tiergarten, Berlin, Germany in SUMO. The simple network simulator of VSimRTI is used for packet level simulations. To study the resilience of our framework in the presence of attackers, we take three attack models: 1) Alteration attack, in which a node alters packet content. If the original packet reports an incident, the attacker will change it to DENM Negation packet which reports nonexistence of a traffic abnormality. 2) Fake information attack, in which an attacker broadcasts a bogus DENM message to its neighbors. This attacker can also collude with other malicious vehicles to affect a bigger area. 3) Message suppression attack, in which an attacker shows selfish behavior and prevents packet propagation.

To evaluate the DTM part of our scheme, we use accuracy as the ratio of correct decision makings to correct plus incorrect decisions. Performance of MND is evaluated by using accuracy, precision, recall, F1, TNR, FPR, and FNR parameters. Result are shown in the next section. Each value on the plots is the average of 300 runs. Weighted voting is used as the baseline method for comparisons. For the sake of comparison, we implement another scheme called RMCV [11]. Other settings used in the simulations are shown in Table II.

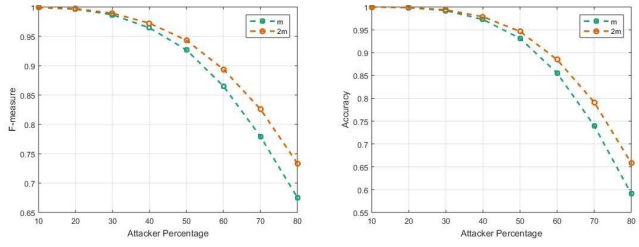
B. Results

Based on Fig. 6, our DTM approach always yields better results than the baseline method. The reason becomes apparent when we compare the ratio of corrupt packets (Fig. 7). In our approach, the attack effect is dampened by the filtering mechanism. Performance of DTM under suppression attack is higher than alteration, as decision boundary is less clear when we have conflicts in information. On the other hand, when the main packets are faked, the extent of invalid information is higher than the other two cases, and thus, even the baseline approach does not show low performance. Altogether, not only



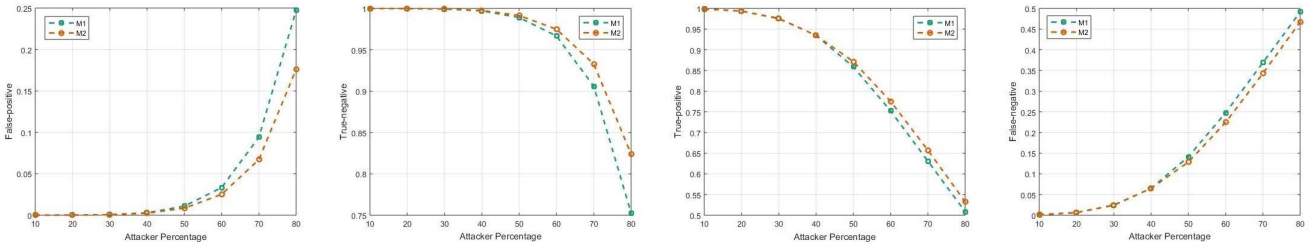
(a) (b)

Fig. 10. (a) The effect of the number of evidences on MND accuracy. (b) The effect of the number of evidences on the F1 measure of MND.



(a) (b)

Fig. 11. (a) The effect of information volume (i.e. m) on MND accuracy. (b) The effect of information volume (m) on the F1 measure of MND.

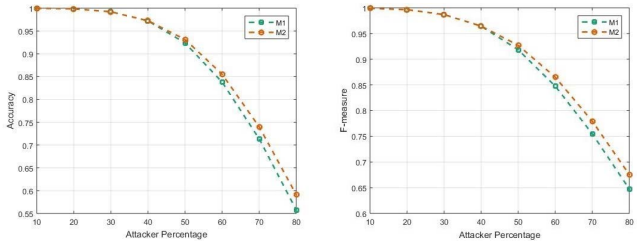


(a) (b) (c) (d)

Fig. 12. Detection performance with unweighted and weighted approaches (a) False positive (b) True negative (c) True positive (d) False negative.

TABLE II
DEFAULT SIMULATION PARAMETERS

Parameter	Value
No. of nodes	50,100
No. of malicious nodes	10%, 20%,. . . , 80%
Transmission range	100 meters
Car following model	Krauss
CAM TTL	10 hops
DENM TTL	15 hops
Loss rate	0.3,0.5, 0.7
Simulation time	400 seconds
Max Num. of nodes in DB	400
Node placement	Random
Max num. of pieces of Inf.(j)	200, 400
No. of evidence (ne)	15, 20, 25



(a) (b)

Fig. 13. Detection performance with weighted (M1) and unweighted (M2) approaches (a) Accuracy (b) F1 measure.

the accuracy of our scheme is higher than the baseline, the decrease in performance is lower than the baseline too.

RMCV approach performs better as the number of attackers increases. In RMCV, a parameter called path similarity is used as the penalty in calculation of trust scores. This parameter assesses the similarity of routing paths for a message in a cluster. As the number of messages and number of relay nodes in the cluster that pertain to the corrupted data increase, the corresponding trust value decreases, and therefore, the accuracy increases. On the other hand when number of attackers is low, the penalty is lower and the trust value representing the corrupt data cluster is higher, even higher than the trust value of trustworthy data, which leads to incorrect decision. Overall, our approach outperfroms RMCV by a large margin, specifically when the ratio of attackers is lower than 50%. In RMCV, when there is one cluster or group of messages regarding an event. The single trust value is compared against a threshold and if its below that threshold, the event report is considered untrustworthy. In our analysis, we use the same threshold for suppression and bogus information attacks, which leads to poor performance in bogus information attack and high, yet lower than DTM, for suppression attack.

Figure 8 shows DTM performance in two networks with different densities. Generally, as the number of nodes increases, the diversity of packet' propagation paths gets higher. On the other hand, at lower node densities, it is more likely that packets get to destination through shorter paths, and with lower chance of modification. When attacker is of selfish type, network size does not affect the performance. In the case of alteration attack, when network size is smaller, accuracy of decisions is higher as more sensors can directly deliver their packets to destinations. Note that in this type of attack, sensor nodes are honest. On the other hand, performance of DTM is better in bigger networks for bogus information attack, where direct messages are corrupted but more packets are expected to be received due to higher number of neighboring nodes. Since CAM messages contain detectable information, the number of correct decisions increases. In addition to the previous experiments, we studied how much the performance of DTM will be affected when network conditions change (see Fig. 9). In suppression attack, higher loss rates translate to lower attacker percentage. Therefore, the overall accuracy increases with the increase of loss. In alteration attacks, accuracy is the highest when loss rate is the lowest, since honest nodes have less chance to broadcast the correct data. When loss increases, less corrupted data samples will be disseminated in the network, so decision is made with higher accuracy.

Next, we test our MND module. According to the results, by increasing the number of attackers, false positive and negative rates increase. This is because more packets will get corrupted, so even honest nodes participate in distributing corrupted packets and thus, the overall trust decreases. This makes the classification boundary narrow as the trust values get close to each other. With the increase in the number of evidences, the difference between the trust values of honest and malicious nodes increases, so the overall performance improves (Fig. 10). A similar increase in MND performance happens when more pieces of information (m) are available (Fig. 11).

In calculating belief and disbelief values, we modeled the contribution of each node in relaying data by a weight. To realize how much node distance from source affects the belief/disbelief value in the model, we compared the basic case with equal weights (Eq. (24) and Eq. (25)) with the weighted

approach of ours (Eq. (26) and Eq. (27)). The results are shown in Fig. 12 and Fig. 13. In these figures, M2 stands for the basic approach and M1 represents the proposed weighted approach. The two methods work almost identically when most of the network nodes are honest, but with an increase of attackers from 50% onwards, our weighted approach excels, with 10% increase in false positive and true negative rates, 5% and 2.5% increase in accuracy and F1 measure, respectively.

V. Conclusion & Future Work

In this paper, a novel scheme, called DTM, is proposed for VANETs that gives nodes environmental cognitive abilities with respect to the event messages routed in the network. It contains two modules, a cognitive trust model to evaluate the trustworthiness of received data pieces, and a malicious node detection module. Our framework utilizes both CAM and DENM data services in VANET to calculate the trust to a received vehicular message. The first module is decentralized, scalable, fast, and accurate. Our results shows that DTM is resilient to different types of adversaries. The malicious node detection module is centralized. Evidences from multiple events are aggregated in a central server or RSU so that deduction about nodes' honesty can be made precisely. This module can classify nodes' nature with an accuracy of 93%, even when half of network is compromised. In future efforts, we strive to estimate these quantities from other available sources so that we can further increase the accuracy.

References

[1] European Commission. (Apr. 2018). *Road Safety in the European Union*. [Online]. Available: http://ec.europa.eu/transport/roadsafety/pdf/vademecum2015.pdf

[2] E. C. Eze, S.-J. Zhang, E.-J. Liu, and J. C. Eze, "Advances in vehicular ad-hoc networks (VANETs): Challenges and road-map for future development," *Int. J. Autom. Comput.*, vol. 13, no. 1, pp. 1–18, Feb. 2016.

[3] N. Toorchi, M. A. Attari, M. S. Haghighi, and Y. Xiang, "A Markov model of safety message broadcasting for vehicular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 1657–1662.

[4] J. Zhang, "Trust management for VANETs: Challenges, desired properties and future directions," *Int. J. Distrib. Syst. Technol.*, vol. 3, no. 1, pp. 48–62, 2012.

[5] C. A. Kerrache, C. T. Calafate, J. Cano, N. Lagraa, and P. Manzoni, "Trust management for vehicular networks: An adversary-oriented overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016.

[6] F. Farivar, M. S. Haghighi, A. Jolfaei, and S. Wen, "On the security of networked control systems in smart vehicle and its adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3824–3831, Jun. 2021.

[7] Z. Huang, S. Ruj, M. A. Cavenaghi, M. Stojmenovic, and A. Nayak, "A social network approach to trust management in VANETs," *Peer Peer Netw. Appl.*, vol. 7, no. 3, pp. 229–242, 2014.

[8] S. Dietzel, J. Petit, G. Heijenk, and F. Kargl, "Graph-based metrics for insider attack detection in VANET multihop data dissemination protocols," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1505–1518, May 2013.

[9] M. S. Haghighi, S. Wen, Y. Xiang, B. Quinn, and W. Zhou, "On the race of worms and patches: Modeling the spread of information in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2854–2865, Dec. 2016.

[10] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *Proc. IEEE INFOCOM 27th Conf. Comput. Commun.*, Apr. 2008, pp. 1238–1246.

[11] S. Gurung, D. Lin, A. Squicciarini, and E. Bertino, "Information-oriented trustworthiness evaluation in vehicular ad-hoc networks," in *Proc. Int. Conf. Netw. Syst. Secur.*, 2013, pp. 94–108.

[12] R. A. Shaikh and A. S. Alzahrani, "Intrusion-aware trust model for vehicular ad hoc networks," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 1652–1669, Nov. 2014.

[13] N.-W. Lo and H.-C. Tsai, "A reputation system for traffic safety event on vehicular ad hoc networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2009, pp. 1–10, Dec. 2009.

[14] Q. Ding, X. Li, M. Jiang, and X. Zhou, "Reputation-based trust model in vehicular ad hoc networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2010, pp. 1–6.

[15] A. Wu, J. Ma, and S. Zhang, "RATE: A RSU-aided scheme for data-centric trust establishment in VANETs," in *Proc. 7th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2011, pp. 1–6.

[16] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicular networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 3, pp. 407–420, May 2011.

[17] F. G. Mármol and G. M. Pérez, "TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 934–941, 2012.

[18] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.

[19] C. A. Kerrache, C. T. Calafate, N. Lagraa, J.-C. Cano, and P. Manzoni, "RITA: RIsk-aware trust-based architecture for collaborative multi-hop vehicular communications," *Secur. Commun. Netw.*, vol. 9, no. 17, pp. 4428–4442, Nov. 2016.

[20] B. Jafarian, N. Yazdani, and M. S. Haghighi, "Discrimination-aware trust management for social Internet of Things," *Comput. Netw.*, vol. 178, Sep. 2020, Art. no. 107254.

[21] M. S. Haghighi, M. Ebrahimi, S. Garg, and A. Jolfaei, "Intelligent trust-based public-key management for IoT by linking edge devices in a fog architecture," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12716–12723, Aug. 2021.

[22] J. Zhang, C. Chen, and R. Cohen, "Trust modeling for message relay control and local action decision making in VANETs," *Secur. Commun. Netw.*, vol. 6, no. 1, pp. 1–14, Jan. 2013.

[23] Y.-C. Wei and Y.-M. Chen, "Reliability and efficiency improvement for trust management model in VANETs," in *Human Centric Technology and Service in Smart Space* (Lecture Notes in Electrical Engineering). Dordrecht, Netherlands: Springer, 2012, pp. 105–112.

[24] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[25] Y. Chen and Y. Wei, "A beacon-based trust management system for enhancing user centric location privacy in VANETs," *J. Commun. Netw.*, vol. 15, no. 2, pp. 153–163, Apr. 2013.

[26] CAMP Vehicle Safety Communications Consortium, "Vehicle safety communications project: Task 3 final report—Identify intelligent vehicle safety applications enabled by DSRC," U.S. Dept. Transp., Tech. Rep. HS-809859, 2005.

[27] M. E. Renda, G. Resta, P. Santi, F. Martelli, and A. Franchini, "IEEE 802.11p VANets: Experimental evaluation of packet inter-reception time," *Comput. Commun.*, vol. 75, pp. 26–38, Feb. 2016.

[28] R. K. Schmidt, R. Lasowski, T. Leinmiiller, C. Linnhoff-Popien, and G. Schafer, "An approach for selective beacon forwarding to improve cooperative awareness," in *Proc. IEEE Veh. Netw. Conf.*, Dec. 2010, pp. 182–188.

[29] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On data-centric misbehavior detection in VANETs," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Sep. 2011, pp. 1–5.

[30] A. Jøsang, *Subjective Logic*. Berlin, Germany: Springer, 2016.

[31] G. A. Korn and T. M. Korn, "Appendix B: B9. Plane and spherical trigonometry: Formulas expressed in terms of the haversine function," in *Mathematical Handbook for Scientists and Engineers: Definitions, theorems, and Formulas for Reference and Review*, 3rd ed. Mineola, Texas: Dover, 2000, pp. 892–893.

[32] A. Jøsang, "The consensus operator for combining beliefs," *Artif. Intell.*, vol. 141, pp. 157–170, Oct. 2002.

[33] R. Protzmann, B. Schünemann, and I. Radusch, "Simulation of convergent networks for intelligent transport systems with VSimRTI," in *Networking Simulation for Intelligent Transportation Systems: High Mobile Wireless Nodes*. Wiley, 2017, ch. 1. [Online]. Available: https://www.wiley.com/en-us/Networking+Simulation+for+Intelligent+Transportation+Systems:+High+Mobile+Wireless+Nodes-p-9781848218536

[34] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.