IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 4, 2021, accepted August 13, 2021, date of publication August 24, 2021, date of current version August 30, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107467

RTEAM: Risk-Based Trust Evaluation Advanced Model for VANETs

RASHA JAMAL ATWA, (Graduate Student Member, IEEE), PAOLA FLOCCHINI, AND AMIYA NAYAK, (Senior Member, IEEE)
School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada
Corresponding author: Amiya Nayak (nayak@uottawa.ca)

ABSTRACT In Vehicular ad hoc networks (VANETs), vehicles share and exchange information regarding road safety and traffic conditions. Thus, trust is established among vehicles to ensure the integrality and reliability of the received reports. Ensuring the security of VANETs is the key to enhance road safety, and for this purpose, several trust establishing, evaluation, and management models have been proposed. When a vehicle receives conflicting reports about an event such as a car accident from its neighboring vehicles, the receiving vehicle must decide which report has to follow. Therefore, the vehicle takes advantage of the available data about the report's sender. Then, the vehicle takes the right action. To this end, we propose a Risk-based Trust Evaluation Advanced Model (RTEAM) based on Multifaceted Trust and Hop-based trust to take action. The proposed model provides a decision-making process according to the risk estimation for each required action of both reports (i.e., reports that deny or confirm the event). The risk is estimated according to the likelihood of taking an incorrect action and its associated impact. Finally, a decision is made corresponding to the action with the lowest risk. The experimental results show that the proposed model shows that the risk-based trust model outperforms a purely trust-based model in terms of undefined cases and true positive rates.

INDEX TERMS Event validation, event detection, trust management, trust establishment, trust evaluation, risk estimation, vehicular ad-hoc networks.

I. INTRODUCTION

In 2016, the estimated number of vehicles in the world was around 1.32 billion vehicles between personal cars, trucks, and buses. This number is expected to reach 2.8 billion vehicles by 2036 [1]. Therefore, the number of accidents is considerably increasing too. Thus, Vehicular Ad-hoc Networks (VANETs) have been proposed as a solution to improve transportation efficiency, ensure road safety, and satisfy road users. In the VANET environment, the vehicles can communicate with each other in ad-manner (V2V), and with the road infrastructure (V2I) through Dedicated Short Range Communication (DSRC) radio [2]. During this communication, the vehicles exchange messages to support safety (e.g., accident warnings) and non-safety (e.g., entertainment) applications. The vehicles cooperate and share some information through periodically Cooperative Awareness Messages (CAMs) (i.e., beacon) [3]. The exchanged messages with the surrounding neighbors and infrastructure may contain information about the sending vehicle itself, such as its location, direction, speed, and control information [4]. Thus, security, privacy, and trust are the main requirements of designing VANETs [5], [6]. Trust is defined as the key element to support security in vehicular networks and describes the level to which a vehicle accepts to rely on another vehicle [7], [8]. In other words, the vehicles[1] have to establish trust among each other and maintain the trust levels during the communication; then, trust is evaluated, the process called the trust management. So, for any received message (M) (i.e., safety or warning message), the receiving vehicle computes the trust level of the sender and/or checks the trustworthiness of the received information. Then, the receiver decides to accept or reject the received message and take the right action regarding that associated information. Thus, it is essential to consider trust metrics since spreading incorrect/inaccurate information through the network may lead to minor or major

The associate editor coordinating the review of this manuscript and approving it for publication was Usama Mir.

[1]In this paper, we used the terms "vehicle", "entity", and "node" interchangeably.

issues on the road and affect the overall traffic safety and network efficiency.

During the last decade, various papers have addressed a number of security threats in VANETs, such as communication attacks (i.e., message forging and tempering) [5], [9]. On the other side, several solutions have been proposed to overcome the VANETs security threats [10]. All of the existing trust models are categorized into three categories based on their trust evaluation mechanisms: 1) Entity-centric, 2) Data-centric, and 3) Combined trust model [11]. The entity-based trust model evaluates the trustworthiness of the entity itself (i.e., the vehicle), the data-based trust model evaluates the trustworthiness of the data reported by the vehicle, and the combined trust model is the combination of the entity-centric and the data-centric models. The existing trust models evaluate trust based on the available information (e.g., detecting the state of the road) and take the appropriate action. However, the risk is not addressed and considered in these existing trust models; this motivates us to propose a risk-based trust evaluation advanced model called *RTEAM*, which identifies the event's occurrence based on the risk of accepting or ignoring the message regarding that event. RTEAM model is an advanced trust model that ensures the trustworthiness of the sender vehicle (i.e., entity-centric model). Once a vehicle receives conflicting reports about an event, RTEAM checks the validity and the relevancy of the reports one by one (i.e., the report should be valid and relative to the receiver's path). Then, RTEAM assesses the authentication of the sender and its trust level as a main security requirement before accepting any information from that sender. Finally, the risk of taking an action (i.e., accepting and rejecting an event) is estimated, and RTEAM takes action with the associated lowest risk. To the best of our knowledge, this is the first paper that uses trust to estimate risk and take action with the lowest risk. We summarize the significant contributions of this paper as follows:

- The paper proposes RTEAM, which is an entity-centric trust model that detects the event state according to the associated risk of believing or disbelieving the occurrence of the reported event.
- RTEAM reduces the processing time, saves resources, and consumes energy by following a two-phase filtering scheme. Phase one filters all invalid and irrelevant reports. Then, in phase two, RTEAM ensures the sender's authentication and trust level before accepting the report.
- The simulation results show that RTEAM outperforms a pure trust-based model in terms of the number of undefined cases and true positive rate.

The remainder of this paper is organized as follows: Section II provides an overview of related work. Section III presents the proposed model in detail. We conduct a set of experiments in Section IV to evaluate the performance of our proposed model. Finally, Section V includes discussion, and Section VI concludes this work and proposes future work.

II. RELATED WORK

In this section, we explore relevant proposed trust models that are currently proposed trust and related to RTEAM. The main purpose of any trust model is to ensure that the communication between nodes is secure (i.e., nodes are trusted and/or data is reliable). Thus, trust management is the key to improve the security and the efficiency of VANET, and also to guarantee the users' (drivers) satisfaction about the provided services. We present the research findings according to the earlier mentioned classification in Section I.

A. ENTITY-CENTRIC TRUST MODELS

Entity-centric trust models (ECTM) evaluate the trustworthiness of the vehicles [12]. The trust model aims to estimate the trust metric according to the node's experience with its neighbors (direct and/or indirect). By estimating the node's trust metric, we can ensure that we can protect the node from malicious nodes. To achieve this, direct interaction is made before deciding to rely on the neighbors' opinions to make a decision. In general, in cluster-based approaches, the elected cluster head (CH) leverages trust computing and/or aggregation such as work in [13], [14]. However, in non-cluster approaches, the node itself is responsible for the trust metric of the targeted neighbor similar to work in [15]–[17]. Minhas *et al.* in [15], proposed a multifaceted trust model that incorporates role-based, direct experience-based, priority, and majority-based trust to detect the event's true state and make a real-time decision. The neighboring vehicles are ordered from the highest to the lowest role/experience-based trust values. Then, when a node seeks advice, it restricts the number of the receivers based on the task on hand (priority); then, the node sends requests to selected neighbors. Once the node receives all responses, majority-consensus is applied to identify the event's true state (i.e., event occurred or not). If the majority consensus is exceeded, then the node accepts the advice; otherwise, it takes the opinion of the receiver with the highest role/experience trust. The main limitation of this approach is that it cannot detect the event's true state if two vehicles with conflicting reports have the same experience/role-based trust. The nodes in Minhas's proposal are responsible for trust evaluation; however, other existing researches employ the static infrastructure (RSUs) to estimate the trust values, distinguish the malicious nodes from the trusted ones, and support communication and trust management framework. RSUs have a higher transmission range and larger storage capacity than the vehicles. Thus, it can see the big picture of the network. Thus, relying on secured RSUs in trust evaluating or establishing process can reduce the communication overhead among the vehicles. Marmol and Perez [16], proposed a trust and reputation infrastructure-based trust model (TRIP) where the static infrastructure (RSUs) are responsible for the trust evaluation process. RSU estimates the reputation score based on experience trust (direct trust), recommendations from node's neighbors, and recommendations from a central authority. Similar work in [17] employs the static infrastructure (RSUs)

to establish trust and oversight nodes' behavior to share it with vehicle when they ask. Furthermore, this model uses direct and indirect trust, recommendation and reputation to estimate the vehicle's trust value. The major drawback of this scheme is that the vehicles may do not have enough time to get the required information about a neighboring vehicle to evaluate its trust.

B. DATA-CENTRIC TRUST MODELS

Data-centric trust model (DCTM) estimates the trustworthiness of the data [12], nothing to do with the entity itself. The DCTM collects information from the network that can assess the vehicle to accurately estimate the trustworthiness of the received data (e.g., information about an event's occurrence). Huang *et al.* [19], proposed a DCTM that uses different weighting for the nodes based on the number of hops from the event. In other words, the weight of nodes from one hop from the event is higher than the weight for those nodes within two hops and more. The proposed model overcomes the oversampling and cascading issue (i.e., some vehicles influence other vehicles' opinions). Ding *et al.* in [20], proposed an event-based reputation model that gives the vehicle different roles (i.e., event reporter, event observer, and event participant) and based on the its role, the vehicle evaluates the trustworthiness of the received message. Moreover, RSU is used to manage the long-term trust for the vehicles that commonly use the same path. However, when the RSU cannot provide a trust value for a vehicle (i.e., vehicle uses the route first time), an event-centric mechanism is applied [21]. Other models use various pieces of information to evaluate the trustworthiness of the message,[2] such as content similarity, content conflict, and routing similarity [22], distance calculation and the vehicle's geolocation [23], and location/time closeness and location/time verification [24]. The main limitation of DCTMs is that the models cannot work if the required information is incomplete or redundant [2].

C. COMBINED TRUST MODELS

Combined trust models (CTM) not only evaluate the trust level of the entity but also compute the trustworthiness of the data [7]. Therefore, the CTM inherits the benefits of ECDM and DCTM [2]. A multi-layers fuzz-logic based model proposed by Soleymani *et al.* [9] that estimates direct experience and the sender's plausibility information (i.e., location and time) are then used to detect malicious nodes and tackle the uncertainty of data in the vehicular network in both Line of Sight (LOS) and Non-Line of Sight (NLOS) cases. Also, authors employ fog nodes to ensure the accuracy level. However, employing fog can be used in earlier stage to minimize time and resources. A beacon-based trust management system (BTM) proposed by Chen and Wei [25] aims to prevent spreading false message via VANET. The entity trust (the message's sender) is constructed from beacon message by finding the cosine similarity between the claimed and the estimated values of the vehicle's position, speed, and direction. Then, the event-based trust is computed where a position and movement verification mechanism is used to verify the event's location and the vehicle's movement (the sender), and the indirect event-based trust is computed in the way that gives low trust values for the message sender if it is a forwarder (not generator). Then, the event reputation is computed, and the composite trust is calculated. Finally, the Dempster-Shafer theory is used for combining the opinions, where Soleymani *et al.* [9] have used the direct experience trust to estimate the trustworthiness of the entity. Chen and Wei [25] have used non-experience-based mechanism to estimate the trustworthiness of the entity based on its plausibility information. Thus, it is important to estimate the sender's trust regardless of the used approach. Clearly, from both [2] and [25], the sender's plausibility information is a useful and significant part for estimating the trustworthiness.

D. RISK IN VANETs

Risk estimation in VANET environment has been studied from different points of view: 1) Security perspective [26]–[28], and 2) Application perspective (i.e., The predicted risk of traffic accident) [29]–[31]. Ren *et al.* [26], proposed a risk assessment model to assess the risk of location privacy in VANET based on an attack tree. The proposed model estimates the possibility of the attacker reaching its attack goal (with leakage of victim's location information) based on the attack cost, technical difficulty, and probability to be discovered. Based on the highest threat probability, the vehicle can predict the possible attack scenario and protect itself from the attack. Another risk assessment framework has been proposed in [27]. The proposed security risk assessment framework is based on a conventional security analysis model and attack tree. The risk assessment uses asset, threat, and vulnerability. The authors in [28] provided a context-based risk assessment sheet that can be used for VANETs, where threats were identified according to mobility. From the application perspective, Fitzgerald and Landfeldt [29], proposed a traffic accident risk mitigation based on the neighbors' factors such as driver, vehicle, and environmental. Each vehicle makes its decision individually. However, later, the authors modified their model in [30] by proposing a model that uses the risk estimates of the surrounding neighbors into account. Similar work has been proposed in [31]; however, the risk was estimated based on the road traffic safety level. The risk was derived based on the sensitivity and type of application. The risk was then measured in a quantitative and qualitative manner by taking into account different contexts such as environmental conditions and the driver's age.

However, the authors suggested that this risk estimation model can be integrated with a trust-based model to enhance the efficiency of the decision-making process in VANET. The idea of integrating the trust model with risk assessment has been applied for work in [18] where the authors applied their context-based risk assessment sheet in [28]. Similar work

[2]In this paper, we used the terms "message", and "report" interchangeably.

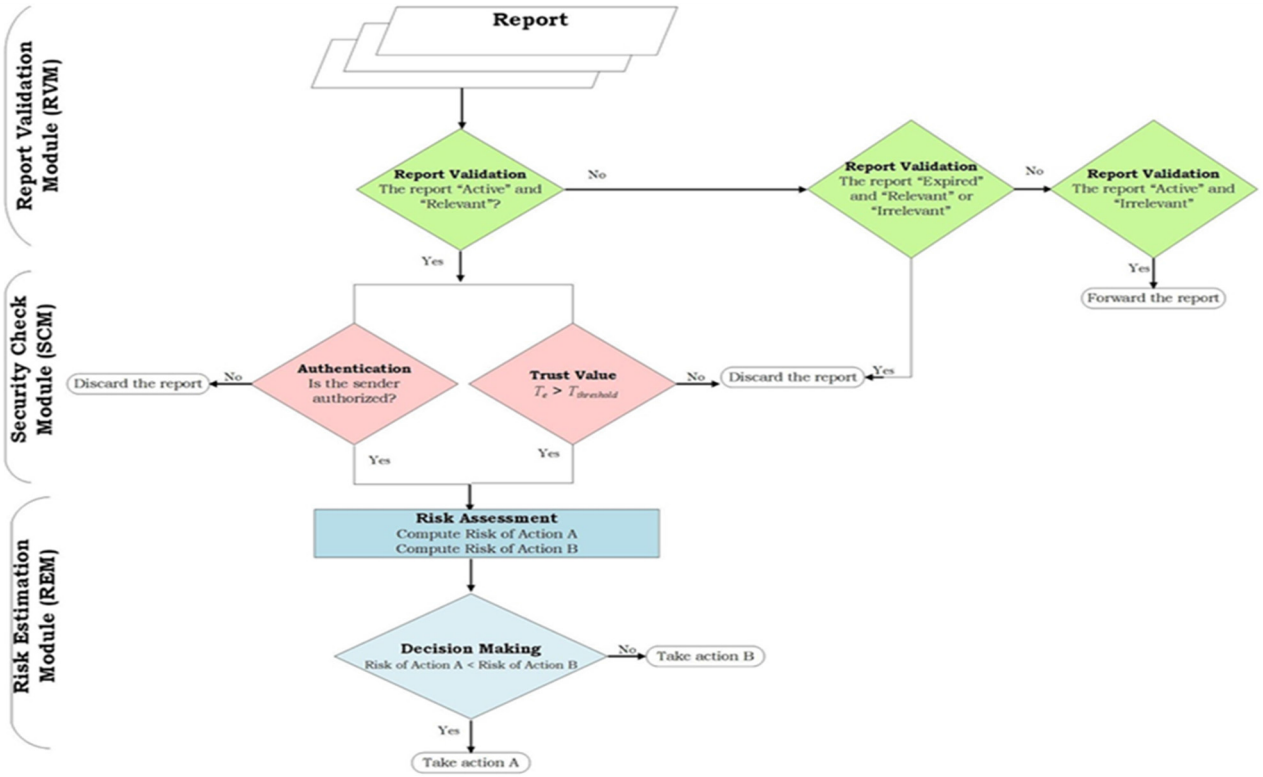

FIGURE 1. RTEAM structure.

has been done in [32], where the proposed security risk assessment was applied to their trust model. However, in [18] and [32], it is not clear enough how the decision can be made based on trust metric and risk level. Thus, we can conclude from the above-mentioned work that risk is estimated separately from trust evaluation, and it can be integrated with the trust management model to enhance the outputs. Thus, the motivation for our work is to take advantage of the trust metric to estimate the possible risk of a taking action regarding an event.

III. PROPOSED MODEL

In this section, we describe our model in details. We propose to develop a risk-based model that aims to identify the event state when a vehicle receiving conflicting reports about the existence of an event. Each vehicle makes a decision to believe or disbelieve the event's occurrence based on the sender's information and the risk of taking the action with the lowest risk. Then, the vehicle spreads its decision (i.e., event or no event) to its neighboring vehicles. The risk-based trust model computes the risk of taking an action based on computing the likelihood and impacts of each action of believing or disbelieving the received report. Four elements are incorporated into our likelihood score computation as follows: 1) Hop-based trust, 2) Experience-based trust (i.e., direct experience), 3) Role-based trust, and 4) Trust consensus. For computing the impacts, two elements are considered as follows: i) the proximity of the vehicle (i.e., the report receiver) to the reported event and ii) the number of neighboring vehicles that may be affected by the vehicle's decision. Then, the vehicle computes the risk of both actions corresponding to believing event or an event and makes a decision to take action with the lowest estimated risk. The proposed model consists of three modules, as depicted in Figure 1: Report Validation Module (RVM), Security Check Module (SCM), and Risk Assessment Module (RAM).

The vehicle receives conflicting reports about an event from neighboring vehicles. First, RVM checks whether the report is still valid and relevant (i.e., affects its path). Then, authentication is checked by using ID authentication (e.g., work in [9]) and the trust level of the sender (i.e., the sender's trust value should exceeded trust threshold), simultaneously. If the sender is authorized and trusted, then, RAM computes the risk of both actions (believing and disbelieving the event) and makes the final decision. In other words, the action with the lowest risk is taken.

A. NETWORK MODEL

First, we give the notations of our network. Let v_i be a vehicle and v_j be a neighboring vehicle of v_i, and v_i, and $v_j \in V$. Each vehicle in the network has a unique identification number (ID) that is issued by a trusted party (e.g., Ministry of Transportation), and a predefined role (T_r), (e.g., a governmental or regular vehicle). Vehicles suppose to have direct communication with its surrounding neighboring vehicles, evaluating the experience and assigning a trust value to each

TABLE 1. Network model notations.

Notations	Description
v_i	Any vehicle in the network, i=1,2, 3,..
v_j	Surrounding neighbor of v_i, j=1,2,3,..
T_r	Role-based Trust of v_j, $T_r \in$ {0.5,0.9}
T_e	Direct Experience-based Trust of v_j, $T_e \in$ [0,1]
Id_j	ID of v_j
L_j	Location of v_j (GPS axis)
S_j	Speed of v_j
D_j	Direct of v_j
N_h	Num. of hops of v_j to the reported event E, N_h=0, 1,..
E^{Id}	Event ID
E^{st}	Event starting time
E^{d}	Event duration in minutes
E^{l}	Event Location (GPS axis)
$t_{curr.}$	Current Time
C_t	Time closeness of v_j between the event's starting time and the time of receiving the report.
C_l	Location closeness of v_j (v_j physical closeness from the event).
$T_{thr.}$	Trust Threshold

neighbor (T_e). At some point, vehicles in V start spreading reports about an event's occurrence on the road. Vehicle v_i receives reports from its preceding neighboring vehicles v_j about an event's occurrence, where a report R indicates the occurrence of the reported event and a report R' negates the occurrence of the reported event. The report of v_j includes the sender's ID (Id_j), the vehicle location (L_j), speed (S_j) and direction (D_j), the number of hops to the reported event (N_h), the event ID (E^{Id}), the event stamp time (E^{st}), the event duration (E^{d}), the event location (E^{l}), and the report context (i.e., R or R'). Vehicle v_i receives the report from v_j at ($t_{curr.}$) and computes time closeness (C_t) of v_j and location closeness (C_l). Note that v_i can only handle report from v_j if v_j is trusted (i.e., meets at least T_{thr}) and report form v_j is valid. We conclude the notations and descriptions in Table 1.

B. REPORT VALIDATION MODULE (RVM)

Due to the high mobility of vehicles in VANET and the size of communication, it is important to allow passing of fresh and related reports. Thus, we propose RVM to provide initial checks to identify the validity of the received report. Vehicle v_i receives a report from neighbor v_j contains information about the sender (e.g., Id_j, L_j, S_j, and D_j) and the event (i.e., E^{Id}, E^{st}, E^{l} and E^{d}). This module is used the same principle in [18], and [9] to identify the validity of the event (i.e., "Active" or "Expired"). Moreover, as in [18], the report relevancy is checked to ensure that the reported event is located in the same city of the receiver. We add the report relevancy check to our report validation module; however, we modified the way of identifying the report relevancy in [18]. In other words, the report relevancy in [18] is checked by ensuring that the sender in the same geographical area (i.e., same city) of the reported event. However, knowing that the event in the city is not enough for the receiver to decide to analyze and handle an event in another part of its city, and it may not be related. This may waste the receiver's resources and time. Thus, we design RVM to check whether the report affects the receiver's path or not. In other words, if the event location is far away from the receiver's location (i.e., does not affect its path), this report is meaningless [33].

When event E is reported for the first time, v_i checks whether or not the event is still active and relevant. For checking the event time validation, the time difference $t_{diff.}$ is calculated between $t_{curr.}$ and E^{st}, then, compared it with E^{d}. In other words, if $t_{diff.}$ is smaller than E^{d}, then, the event is "Active"; otherwise, the event is "Expired". For checking the event relevancy, if the event location E^{l} on the receiver's path, then, event E is "Relevant"; otherwise, the event E is "Irrelevant". The four possible cases that could happen are shown in Table 2. We construct two lists, called EL_{Valid}, $EL_{Invalid}$ to keep track of any future reports regarding the same event. In other words, after E^{d} expires, the event is automatically moved from EL_{Valid} to $EL_{Invalid}$ as invalid event.

It is important to note the following: 1) the event duration E^{d} depends on the event type [9], [18]. For a major event such as road closure due to an accident or construction, E^{d} is about 60 to 120 minutes, while for a minor event such as a minor traffic accident, E^{d} is about 30 to 40 minutes [18], 2) according to our assumption, the receiver has to receive at least two conflicting reports, 3) the case of receiving a single report or similar reports about an event is not in the scope in this work, 4) any report that follows *Case 1* (see Table 2) is added to EL_{valid} regardless of its content (i.e., agree with the event or not), 5) the RVM prevents spreading invalid reports through the network and saving the vehicle's resources and time, 6) any vehicle that keeps sending or forwarding invalid reports should be reported to the road monitor (e.g., RSU).

C. SECURITY CHECK MODULE (SCM)

This module assesses the authentication of the sender and the sender's trust level, which is a basic step before going further. In other words, the sender should be a trusted VANET participant (i.e., v_j is a trusted participant). Like RVM, SCM mainly aims to save the resources and the time of the receiver from unauthorized and distrust sender. For authentication purpose, we suggest using the authentication scheme proposed in [6] to verify the authenticity, where the vehicles use a certificate issued and revoked by a Certificate Authority (CA). The authentication check protects the vehicles in the network from cybersecurity [34] and prevents unreliable/fake senders from spreading their reports via the network. Simultaneously, the sender's trust level is checked, and it should be greater than the trust threshold (e.g., $T_{thr.} > 0.6$, [35]). As shown in Figure 1, the report from the sender that does not meet the authentication check or the trust level condition is discarded. Otherwise, the report is accepted. Then, after n reports (n at least two conflicting reports), the risk estimation is calculated.

TABLE 2. RVM possible cases and the required action.

Case Num.	*Description*	*Action*
Case 1	**Report "Active" and "Relevant"**	First time received a report about E: **Add to** EL_{valid} n^{th} time received a report about the same E: **Move to the next step**
Case 2	**Report "Active" and "Irrelevant"**	**Forward the report to the vehicles that may affect by the event (e.g., the vehicles in the opposite direction).**
Case 3	**Report "Expired" and "Relevant"**	**Discard the report** **Move the report to** $EL_{invalid}$
Case 4	**Report"Expired" and "Irrelevant"**	**Discard the report**

D. RISK ASSESSMENT MODULE (RAM)

By reaching this point, the receiver v_i received nth contradictory reports (R and R') about an event's occurrence E from its neighboring vehicles (i.e., report R informs that event E does exist, and report R' informs that event E does not exist). RAM provides a decision-making process for vehicles facing conflicting reports. This is done by aggregating nth reports from vehicles within v_i transmission range, then based on certain metrics, v_i decides on the existence of an event before propagating its final decision to other neighbors behind it. Vehicle v_i computes the risks for the associated action of believing R or R', then, the it makes a decision to take action with the lowest estimated risk. This module comprises two phases: 1) Risk assessment phase and 2) Decision-making phase.

1) PHASE 1: RISK ASSESSMENT

In this phase, risk estimation is integrated with a multi-dimensional trust model that uses experience-based, role-based, hop-based and majority-based approach. We assume that each vehicle has a direct communication (T_e) with its neighbors (i.e., at least one interaction). We focus on risk estimation. However, for computing and updating the direct trust, we suggest using the trust model proposed in [15]. Risk is defined in accordance with the USA National Institute of Standards and Technology (NIST) in [36], as follows:

$$Risk = Likelihood \times Impact \tag{1}$$

According to our work, we define the *Likelihood* as the probability of making an incorrect decision in the face of conflicting reports, whereas the *Impact* is defined as the consequence of that incorrect decision.

Suppose that vehicle v_i receives nth conflicting reports (R) and (R'). The true state of an event is θ may then be one of two possibilities: 1) θ_R: the event did occur (i.e., report R is true), or 2) $\theta_{R'}$: the event did not occur (i.e., report R' is true). If θ_R is believed, the vehicle takes an action a_R, whereas if $\theta_{R'}$ is believed, the vehicle takes an action $a_{R'}$. In both cases, the vehicle notifies the other drivers of its decision regarding the event. Clearly, the objective is to take the action a_R when, in fact, $\theta = \theta_R$, and the action $a_{R'}$ when $\theta = \theta_{R'}$. Thus, we have a binary set of states $\theta = \{\theta_R, \theta_{R'}\}$ and a binary set of actions A $= \{a_R, a_{R'}\}$, representing a scenario that can be interpreted as a hypothesis on θ, where action a_R is taken if the hypothesis θ_R is believed. Two types of errors may therefore be committed in this situation as follows: 1) A *Type I* error is to take the action a_R when $\theta = a_{R'}$. This error results in a false notification to the drivers of an event's occurrence. On the other hand, a *Type II* error is to take the action $a_{R'}$ when $\theta = \theta_R$. This error results in a false notification to the vehicles, unintentionally misleading the drivers despite the occurrence of an event. For simplification, we called a *Type I* error (*ERR*1) and a *Type II* error (*ERR*2). Using Eq. (1), a separate risk function L can be defined for each action as:

$$L(a_R, \theta) = \alpha(\theta|a_R) M_\alpha \tag{2}$$

$$L(a_{R'}, \theta) = \beta(\theta|a_{R'}) M_\beta \tag{3}$$

where $\alpha(\theta|a_R)$, and $\beta(\theta|a_{R'})$, represents the *likelihood* of a *ERR*1, given action a_R; and the *likelihood* of *ERR*2, given action $a_{R'}$, and M_α and M_β are the *impacts* associated with the *ERR*1 and *ERR*2 errors, respectively; In the following, we shorten the notation $\alpha(\theta|a_R)$ and $\beta(\theta|a_{R'})$ to α and β for convenience.

Before explaining likelihood score computation, we give a brief description of the main components of the likelihood computation formula. We use the same category in [21], where the vehicles are divided based on their relation with the event in three categories: 1) Event Reporter (the vehicle involved in the event), 2) Event Observer (the vehicle witnesses the event and within one hop from the event reporter), and 3) Event Participant (the vehicle is within two or more hops away from the event report). For simplicity, we use the abbreviation for each category as ER, EO, and EP, respectively. The likelihood score formula combines the work in [15] and the weighting scheme in [19]. The weighting scheme uses a hop-based model that aims to overcome cascading and oversampling issue by giving the highest weight for the first observer (i.e., EO) and the lowest weight for vehicle two or more hops from the event (i.e., EP). The model uses hop-based trust to detect whether an event exists or not. The hop weight ($\alpha_{hop=1,2,\ldots,n}$) is multiplied by the vehicle decision (i.e., $d_j = 1$ if v agrees with the event; otherwise, $d_j = -1$). Then, the hop weights by the corresponding decision are aggregated (W_d). The event exists if W_d is greater than 0; otherwise, it does not exist. The model does not show

the case when W_d is exactly equal to 0. Also, the report generator (i.e., ER) is neglected in [19]. Therefore, we use the same concept of hop weighting scheme in [19], but we consider ER in our weighting scheme.

For the likelihood score computation, we add a hop-based trust metric to trust consensus, as proxies from α and β. We update the formula of calculating the aggregated effect for report R_j from vehicle v_j in [37] by integrating the weight of each report based on the number of hops from the event, as follows:

$$E\left(R_j\right) = \sum\nolimits_{v_j \in V} W(R_j) \left[\frac{T_e\left(v_j\right) T_r(v_j)}{C_t\left(v_j\right) C_l(v_j)} \right] \tag{4}$$

where $E(R_j)$ is the aggregated effect formulae for report R_j, T_e is the experience-based trust factor, and T_r is the role-based trust factor, C_t and C_l are the time closeness and location closeness, respectively, and $W(R_j)$ is the weight of report R_j based on the number of hops from the event, which can be expressed as follows:

$$W\left(R_j\right) = \begin{cases} \omega & if\ hop = 0\ (v_j\ is\ ER) \\ \omega - 1 & if\ hop = 1\ (v_j\ is\ EO) \\ (\omega - 2)^{\frac{1}{hop}} & if\ hop\ \geq 2\ (v_j\ is\ EP) \end{cases} \tag{5}$$

where the constant $\omega > 2$.

We divide the vehicles into two sets according to their reports regarding the event's occurrence and then the aggregated effect of the reports of both sets $E(R)$, and $E(R')$ are computed using Eq. (4). Then, the *likelihood* score is computed as proxies for α and β as:

$$\alpha = 1 - \beta = \frac{E\left(R'\right)}{E\left(R\right) + E\left(R'\right)} \tag{6}$$

Note that $E(R)$ and $E(R')$ are non-negative. Since our goal is optimization (i.e., to take action associated with the lowest risk), it is the relative values of α and β that are of importance.

Moving to compute the impacts of incorrect action, first we defined the impacts M_α and M_β as the consequences of an incorrect action due to the existence of *ERR*1 and *ERR*2 errors. We defined a factor called the error intensity I that presents the measure of the vehicle damage size due to *ERR*1, and *ERR*2 errors, respectively. Also, we say that the impacts (M_α and M_β) are proportional to the error intensity (I_α and I_β), and we model the impacts, as follows:

$$\frac{M_\alpha}{M_\beta} = \frac{I_\alpha}{I_\beta} \tag{7}$$

where M_α/M_β is the risk ratio.

As earlier mentioned, *ERR*1 commits in the situation where the driver gets a false notification about an event and takes action a_R; however, the true state of the event is $\theta_{R'}$. The driver's action a_R could be slowing down the speed, changing lane, or entering the nearest exit. The type of the action a_R and *ERR*1 error intensity (I_α) is dependent on the event type and the proximity of the vehicle to the reported event. Let vehicle v_i be on the highway, and the driver receives a notification about an accident within T time from its location, and there are N neighbors following the vehicle (i.e., may be affected by its outgoing report). The estimated T to the purported accident determines the driver's action. For example, if T is high (i.e., the driver would probably drive more slowly) and I_α is close to nil. However, if T is low (i.e., the driver would be alarmed), the drivers are alarmed and take extreme action such as hard braking). The immediate consequence of the *ERR*1 error is having congestion on a lane or on the road due to slowing down the speed. Therefore, the smaller T is, the more disruptive *ERR*1 error is. On the other hand, the larger N is the larger error intensity is. Thus, we model the error intensity I_α of *ERR*1 as:

$$I_\alpha = a + \left({}^{N}/_{T}\right)^{b} \tag{8}$$

where a is the baseline of the error intensity and b is a parameter that adjusts the scale and the shape of the function according to our perception of how I_α changes with T and N.

Now consider the case of *ERR*2 error where there is an actual event (i.e., accident) that has occurred ahead on the highway, but the driver takes action $a_{R'}$. Here, the driver may come upon the event without warning, potentially being forced to brake suddenly or swerve. Thus, the immediate consequence of the *ERR*2 error is to delay the possibility of taking the correct action, a_R. With large T (i.e., 5 min), the driver has ample time to get more reports regarding the event and identify the true event state, then, take the right action a_R. However, with a small T, the driver becomes too close to the accident, and no action is taken, then, this increases the possibility of a major accident occurs on the highway due to late and surprising action by the driver. Even if the correct decision is finally made, a smaller T demands a more abrupt response by the driver. Also, with increasing N neighboring, the impact of *ERR*2 increases too (i.e., more vehicles are affected). Thus, we also interpret I_β to have a form similar to I_α, as follows:

$$I_\beta = c + \left({}^{N}/_{T}\right)^{d} \tag{9}$$

Similar to Eq. (8), c here is the baseline of the error intensity, and d is a parameter to adjust the scale and shape of the function where $d > b$. Note that both T in Eq. (8) and Eq. (9) should be larger than T point at which the event state is truly identified by the vehicle/driver (i.e., by the vehicle's sensors or by the driver's biological sense), we called this point (T_{truth}). At T_{truth}, the driver will take action based on the real state of the event without considering any other computations. Finally, the estimated risks L (a_R, θ) and L ($a_{R'}$, θ) of both actions are calculated using Eq. (2) and (3).

Note that: 1) the number of neighbors following the vehicle N is directly proportional to I, and T is inversely proportional to I, and 2) the shape of *ERR*1 and *ERR*2 intensity curves could take different shapes according to the road conditions (i.e., different scenarios). In the following subsection, we explain the three possible curves of *ERR*1 and *ERR*2 intensity. Also, the scenario mentioned above is one of these scenarios, called RTEAM-2.

2) PHASE 2: DECISION-MAKING

In this phase, the vehicle has to take action a that corresponds to the lowest of risks $L(a_R, \theta)$ and $L(a_{R'}, \theta)$. Thus, the Bayes' decision rule is used [38] to take action with the lowest risk regardless of the correctness of the action. The vehicle takes action a_R, if $\alpha M_\alpha < \beta M_\beta$; otherwise, action $a_{R'}$ is taken. In other words, if the risk ratio $\frac{M_\alpha}{M_\beta}$ is smaller than $\frac{\beta}{\alpha}$, then action a_R is considered; otherwise, action $a_{R'}$ is taken.

Note that for any accepted report that gives the true event state, the sender's trust value is updated (i.e., trust value is increased) [15], [39] and the new $T_{e(j)}$ is updated as follows:

$$T_{e(j)} = \begin{cases} \lambda^t (1-\alpha) T_{e(j)} + \alpha & if\ T_{e(j)} \geq T_{thr}. \\ \lambda^{-t} (1-\alpha) T_{e(j)} + \alpha & if\ T_{e(j)} < T_{thr}. \end{cases} \quad (10)$$

where λ is forgetting factor (to give less weight to older interactions) and $0 < \lambda < 1$, and α is reward factor, and its value is $0 < \alpha < 1$.

On the other hand, for any report that is discarded due to its content (i.e., report with false event state or an expire event), the sender's trust value is updated (i.e., trust value is decreased) [15], [39] and the new $T_{e(j)}$ is updated, where the overall trust of the honest vehicle is computed by Eq. (10) and the overall trust of the malicious vehicle is computed by Eq. (11) as follows:

$$T_{e(j)} = \begin{cases} \lambda^t (1-\beta) T_{e(j)} + \beta & if\ T_{e(j)} \geq T_{thr}. \\ \lambda^{-t} (1-\beta) T_{e(j)} + \beta & if\ T_{e(j)} < T_{thr}. \end{cases} \quad (11)$$

where α, β, λ are reward, penalty, and forgetting factors, respectively. Their values are in the range $0 < \alpha, \beta, \lambda < 1$.

E. IMPACTS SCENARIOS

Urban, rural, and freeway areas differ in road conditions such as traffic volume, allowed speed limit, obstacles (e.g., pedestrians, bicyclists, and school zones), and accident rates. Not just that, but even in the same area, the driver may experience different road conditions during the day (e.g., high volume traffic in the rush hour). Thus, the driver has to be able to apply several techniques and skills to maintain safe driving in each area based on its road conditions. For example, the driver in the urban area is prepared to stop or slow down suddenly, where there is a high possibility of unexpected event occurs (e.g., pedestrians approach the road suddenly); however, the driver in a freeway area (i.e., highway) is not prepared to slow down or cover the brake suddenly. Based on the above mentioned, we expect different realistic scenarios of the impacts of *ERR*1 and *ERR*2 errors under different road conditions. Since we do not have data from the real world to shape the error intensity curves, we assume three possible impact curves of *ERR*1, and *ERR*2 errors, as shown in Figure 2. The impact of both types of errors is increased by time. However, the impact of *ERR*2 is always larger than the impact of *ERR*1 when the vehicle is too close to the event, and no action is taken yet.

- RTEAM-1 (Scenario A) is shown in Figure 2a, where *ERR*2 error is always higher impact than *ERR*1 error, and this an expected scenario in a highway where there is a big chance of a multi-crashes accident if one vehicle makes incorrect action (i.e., switching the lane suddenly). In other words, this scenario reflects the case where the intensity of *ERR*2 (i.e., impact) is always larger than the intensity of *ERR*2, and both impacts are increased the time.
- RTEAM-2 (Scenario B) is shown in Figure 2b, where the intensity of *ERR*2 error (the dashed red curve) is found to be lower than the *ERR*1 (the dashed green curve) at high T because, with *ERR*2 error, the drivers just keep carrying on without disruption. The drivers have time to make the right decision. Somewhere around T = 20sec, the *ERR*2 error starts to become a bigger problem than the *ERR*1 error (i.e., there is an accident on the highway, and the drivers are getting close to it). Note that the point shows how crucial the decision errors become as T gets smaller. This scenario could happen in highways, urban areas with high density, or urban areas in rush hours. In other words, this scenario reflects the case where the intensity of *ERR*1 (i.e., impact) is larger than the intensity of *ERR*2 at high T, and when T is decreased (i.e., the event is close), *ERR*2 is gradually increased until reaching some point where the curve jumps (i.e., the event is too close).
- RTEAM-3 (Scenario C) is shown in Figure 2c, where both errors have the same impact at high T. Then, by the time passing, *ERR*2 impact increases. This scenario could happen where the driver is driving with caution (i.e., aware of any unexpected obstacle/event) in the urban area. In other words, this scenario reflects the case where the impact of both errors is the same when T is high, but over time, *ERR*2 impact gradually increases.

IV. PERFORMANCE EVALUATION

In this section, we use MATLAB to evaluate our proposed model in detail through simulations. The vehicles are set to be on a 3-lane highway, with one lane fully occupied and two semi-occupied ones. The distances between vehicles on the fully occupied lane vary around their 2-second safe distances (e.g., around 44.4 m for an 80 km/hr speed). In each simulation run, an accident is set to occur on the highway and notifications are sent to all vehicles informing them about the event. Reports from event reporters are set to be trusted since they are directly sent by the equipped sensors without human intervention. Vehicles within 100m from the event (i.e., the assumed range of V2V communications [40]) can therefore make decisions based on the received message from the event reporters without using the mathematical model to make a decision. We assume that all reports are valid and all sender's trust values exceed the trust threshold. For our simulations, we vary the percentage of malicious nodes, which tends to negate that any accident ever took place. For our risk-based calculations, we assume that substantial data has been collected about the outcomes of *ERR*1, and *ERR*2 errors with respect to highway accidents,

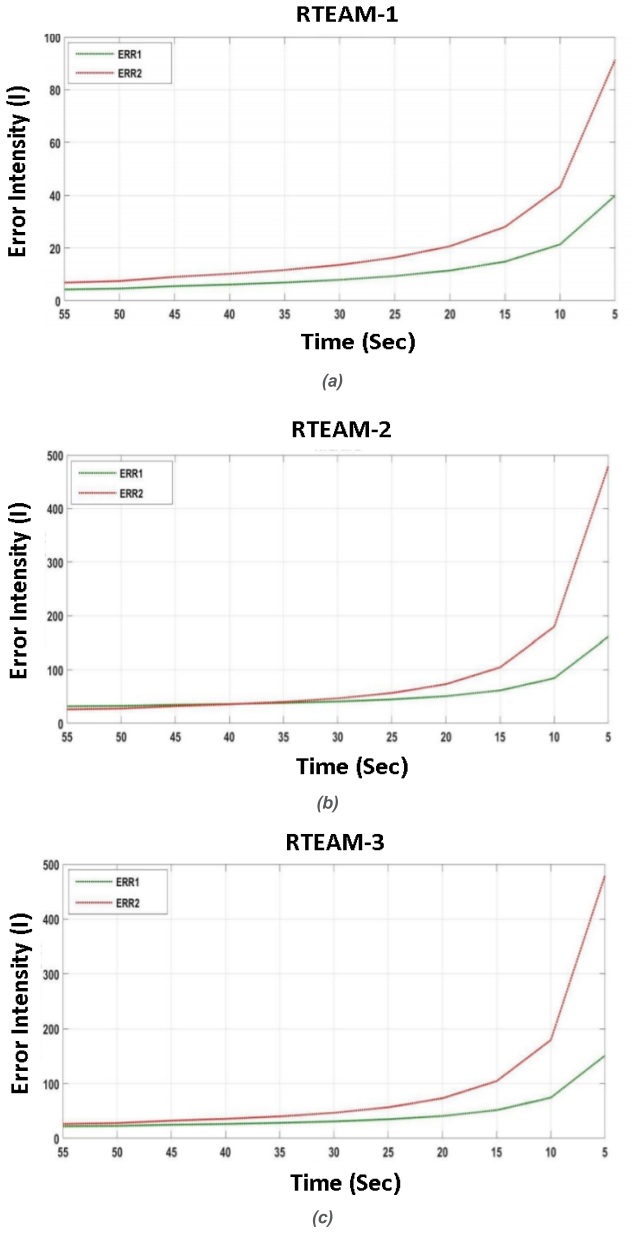

FIGURE 2. ERR1 and ERR2 curves of RTEAM-1,2, and 3.

resulting in the error intensity curves of Eq. (8) and (9) to be defined according to the values of a, b, c, and d as shown in Table 3.

A. PERFORMANCE METRICS

To show the effectiveness of RTEAM, we compare RTEAM with 1) Simple Trust Model (STM) [33], where the earliest report is followed, 2) Hop-based Trust Model (HTM) [19], and Multifaceted Trust Model (MTM) [15].

We defined the following metrics to evaluate the efficiency of RTEAM-1, RTEAM-2, and RTEAM-3 for comparison:

• *Undefined Cases (UND):* This metric reflects the number of cases that a vehicle failed to determine the event state (i.e., whether or not there is an event).

• *True Positive Rate (TPR):* Represents the probability of correctly detecting the event state θ.

B. EVALUATION SCENARIOS AND RESULTS

1) UNDEFINED CASES (UND)

The effective trust model should be able to determine the event state under any conditions. The case where a vehicle could not determine the event state (i.e., whether or not there is an event) is unacceptable. Figure 3 depicts where the MTM fails to define about 3% of the cases when the percentage of malicious vehicles in the network falls between 5% to 15%. The number of undefined cases then slightly decreases as the number of malicious vehicles further increases.

This is because increasing the number of malicious vehicles in the network leads to more cases where the event state is defined (even if it is the wrong one) and fewer cases where it is undefined. However, the worst UND is shown with HTM, where UND is about 20% with less percentage of malicious 5%. The UND is dramatically increased up to 40% until reaching more than 50% when the percentage of malicious got increased from 10% to 25%. The rapid increase in UND cases is because HTM relies mostly on the opinion of the first-hand observers (i.e., one hop from the event) regardless of their trustworthiness. In other words, the weight of the first-hand observers' opinion is the decision in the way that the vehicle cannot identify the event state (i.e., the weight of the vehicles agree with the occurrence of the event minus the weight of the vehicles disagree with the occurrence of an event is equal to zero). With the increase in the number of malicious vehicles in the network (i.e., most of the participants deny the occurrence's of the event), the UND cases decreases (i.e., the decision is mostly unified "No Event").

On the other hand, STM can always determine the event state because it makes its decision based on the earliest received report. However, this method lacks accuracy because the driver makes its decision based on one received report only, and this report may be a fake one. However, RTEAM considers different aspects of the senders and relies on multiple reports before deciding on the action regarding an event state. With respect to MTM and HTM, it can be seen that RTEAM-1, 2, and 3 outperform both models by being able to make a decision in all cases (i.e., all cases are defined) regardless of the percentage of malicious vehicles.

2) TRUE POSITIVE RATE (TPR)

The TPR depicts the probability of correctly detecting the event state θ. Generally speaking, increasing the number of malicious vehicles leads to decreasing the chance of correctly detecting the event state. Increasing the number of attackers gives a high chance for the false event state to spread across the network, which negatively affects the other vehicles' decisions.

As depicted in Figure 4, the TPR of STM, HTM, and MTM emphasizes that the network achieves lower TPR compared to RTEAM-1, 2 and 3. HTM and MTM show slightly similar

TABLE 3. Simulation parameters.

Parameter		Value
Simulation scenario		3-lane highway
Average vehicle speed		80 km/hr
Total number of vehicles (N_v)		100
Number of government vehicles		10%
Vehicle transmission range		100 m
Malicious vehicle percentage		5 – 50%
Vehicle role (T_r)		0.9 for government vehicles and 0.5 otherwise
Vehicle trust (T_e)		0.5
Vehicle time closeness (C_t)		(0, 1)
Vehicle location closeness (C_l)		(0, 1)
Attacker model		non-government vehicles lying about the existence of an event
RTEAM-1	Error baseline of I_α and I_β (a, c)	0, 1
	Scaling parameters (b, d)	0.9, 1.1
RTEAM-2	Error baseline of I_α and I_β (a, c)	25, 15
	Scaling parameters (b, d)	1.2, 1.5
RTEAM-3	Error baseline of I_α and I_β (a, c)	15, 15
	Scaling parameters (b, d)	1.2, 1.5

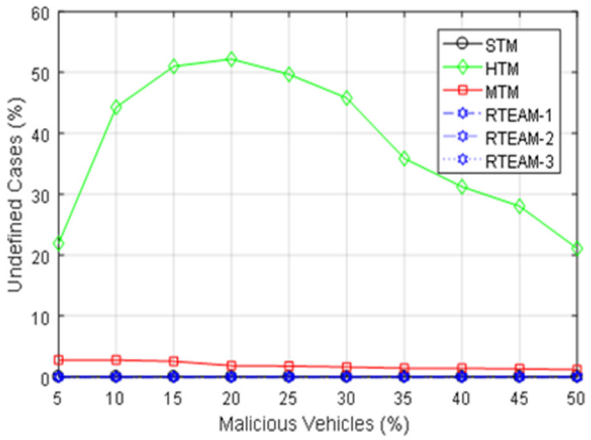

FIGURE 3. Undefined cases (UND).

TPR for different malicious cases. With a network injected with 5% malicious vehicles, RTEAM achieves higher TRP (about 88%) compared to HTM and MTM, where the TRP is less than 80%. This due to the fact that RTEAM inherits the advantage of HTM, and MTM (i.e., Multifaceted trust and hop-based trust are incorporated in RTEAM). The gap between RTEAM-3 and HTM and MTM is about 20%, and it is 30% to 40% higher when the number of malicious

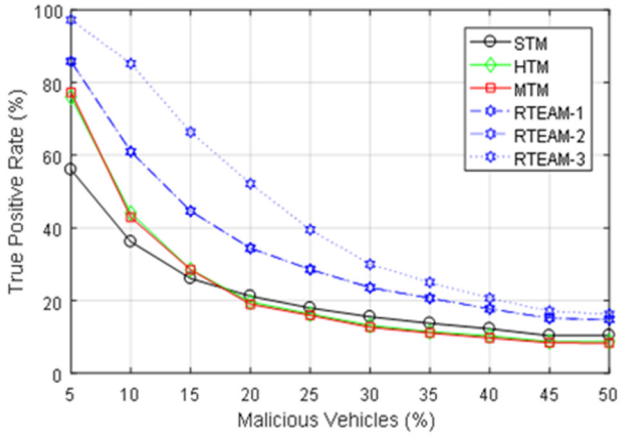

FIGURE 4. True positive rate (TPR).

vehicles is between 10% and 25%. Moreover, RTEAM-3 can achieve higher TPR than STM. RTEAM-3 outperforms all trust models including RTEAM-1 and 2.

RTEAM-1 and 2 show the exact TPR results, and they show better results than other trust models. In short, Figure 4 clearly depicts that RTEAM can achieve higher TPR and get better results than other trust models, especially, when the number of injected attackers is under 30%. Even though RTEAM shows low TPR when the percentage of malicious vehicles is increased (due to the fact that HTM and MTM are parts of its architecture), it still outperforms the other trust models in terms of TPR and can truly detect about 19% of the event state compared to 10% in other trust models where 50% of the vehicles in the network are malicious ones.

V. DISCUSSION

In this section, we discuss RTEAM in terms of limitations and possible improvements. Experimental results show that "Risk" can drive "Trust". In other words, the associated risk of action determines which opinion the vehicle has to follow (i.e., has to trust its advice). Also, RTEAM can work in both clustered and non-clustered networks. Cluster Head (CH) can make the decision (i.e., believing or disbelieving the occurrence's of an event) and send the right advice to the cluster members instead of each vehicle making the decision by itself. Moreover, the unified decision that is made by CH may reduce the risk level and workload on the individual vehicle. One of the main limitations of RTEAM is the design that is based on trust metrics to compute the likelihood, which affects its performance. Another limitation is that RTEAM cannot process the case where we only have a single received report. This limitation can be fixed by adding a submodule to support the decision-making process, such as an infrastructure-based trust evaluation module, where the road infrastructure can help the vehicle to decide the event state.

Other possible modifications can improve the performance of RTEAM are the following: 1) adding a data-based trust evaluation module that checks the correctness of the received

data (i.e., by evaluating the plausibility of the sender), and 2) adding a Payment Punishment Scheme (PPS) that can encourage vehicles to participate in voting on the events in the network. The computation and communication complexity of RTEAM is relatively low due to the fact that it applies a two-phase filtering scheme. Only valid and relevant reports from trusted and authorized senders are accepted and processed, which reduces processing time and saves resources.

VI. CONCLUSION

In this paper, we have proposed a risk-based trust model for VANET. The proposed model improves the decision-making process by integrating risk estimation into the trust evaluation process of incoming reports. Simulation results demonstrated how the risk-based model outperforms a pure trust-based model. This is because the risk-based trust model always seeks the lowest-risk action, whereas the trust-based model decides upon actions based only on the highest trust value reports. This work, therefore, opens the door for many future extensions of this work as follows. The way of calculating the risk impact may be improved by considering the cluster vulnerability, which could then enhance the risk estimation. Exploring different ways to derive the likelihoods (α and β) would be helpful to develop a better understanding of the error intensity curves. Finally, this research may be expanded through the implementation more comprehensive simulations using different scenarios in addition to comparison against other existing models. Also, to overcome the short communication time issue in VANET, this model can be enhanced by utilizing the concept of public trust value for the vehicle that RSUs could provide it upon request.

ACKNOWLEDGMENT

This article was presented at the IEEE International Symposium on Networks, Computers and Communications (ISNCC), Montreal, Canada, in October 2020, [DOI: 10.1109/ISNCC49221.2020.9297329].

REFERENCES

[1] Cars Guide. (Sep. 2020). *How Many Cars are There in the World*. [Online]. Available: https://www.carsguide.com.au/car-advice/how-many-cars-are-there-in-the-world-70629

[2] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 2, pp. 760–776, Feb. 2018.

[3] N. Lyamin, A. Vinel, M. Jonsson, and B. Bellalta, "Cooperative awareness in VANETs: On ETSI EN 302 637-2 performance," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 17–28, Jan. 2018.

[4] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," *Veh. Commun.*, vol. 1, no. 4, pp. 214–225, Oct. 2014.

[5] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "BARS: A blockchain-based anonymous reputation system for trust management in VANETs," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Aug. 2018, pp. 98–103.

[6] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[7] M. Monir and A. Abdel-Hamid, "A categorized trust-based message reporting scheme for VANETs," in *Proc. Int. Conf. Secur. Inf. Commun. Netw.*, 2013, pp. 65–83.

[8] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Depend. Sec. Comput.*, vol. 1, no. 1, pp. 11–33, Jan./Mar. 2004.

[9] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. K. Khan, and S. Goudarzi, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15619–15629, 2017.

[10] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Comput. Commun.*, vol. 44, pp. 1–13, May 2014.

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

[12] S. A. Soleymani, A. H. Abdullah, W. H. Hassan, M. H. Anisi, S. Goudarzi, and M. A. R. Baee, "Trust management in vehicular ad hoc network: A systematic review," *EURASIP J. Wireless Commun. Netw.*, vol. 1, p. 146, Dec. 2015.

[13] U. Khan, S. Agrawal, and S. Silakari, "Detection of malicious nodes (DMN) in vehicular ad-hoc networks," *Procedia Comput. Sci.*, vol. 46, no. 9, pp. 965–972, 2015.

[14] A. Jesudoss, S. K. Raja, and A. Sulaiman, "Stimulating truth-telling and cooperation among nodes in VANETs through payment and punishment scheme," *Ad Hoc Netw.*, vol. 24, pp. 250–263, Jan. 2015.

[15] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "Towards expanded trust management for agents in vehicular ad-hoc networks," *Int. J. Comput. Intell., Theory Pract.*, vol. 5, no. 1, pp. 3–15, Jan. 2010.

[16] 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.

[17] C. A. Kerrache, N. Lagraa, C. T. Calafate, J. C. Cano, and P. Manzoni, "T-VNets: A novel trust architecture for vehicular networks using the standardized messaging services of ETSI ITS," *Comput. Commun.*, vol. 93, pp. 68–83, Nov. 2016.

[18] F. Ahmad, V. N. L. Franqueira, and A. Adnane, "TEAM: A trust evaluation and management framework in context-enabled vehicular ad-hoc networks," *IEEE Access*, vol. 6, pp. 28643–28660, 2018.

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

[20] Q. Ding, X. Li, M. Jiang, and X. Zhou, "Reputation management in vehicular ad hoc networks," in *Proc. Int. Conf. Multimedia Technol.*, Oct. 2010, pp. 1–5.

[21] Q. Ding, X. Li, M. Jiang, and X. Zhou, "A novel reputation management framework for vehicular ad hoc networks," *Int. J. Multimedia Technol.*, vol. 3, no. 2, pp. 62–66, 2013.

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

[23] D. B. Rawat, G. Yan, B. B. Bista, and M. C. Weigle, "Trust on the security of wireless vehicular ad-hoc networking," *Ad Hoc Sensor Wireless Netw.*, vol. 24, nos. 3–4, pp. 283–305, 2015.

[24] 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.

[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] D. Ren, S. Du, and H. Zhu, "A novel attack tree based risk assessment approach for location privacy preservation in the VANETs," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2011, pp. 1–5.

[27] H.-K. Kong, T.-S. Kim, and M.-K. Hong, "A security risk assessment framework for smart car," in *Proc. 10th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (IMIS)*, Jul. 2016, pp. 102–108.

[28] F. Ahmad and A. Adnane, "A novel context-based risk assessment approach in vehicular networks," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Mar. 2016, pp. 466–474.

[29] E. Fitzgerald and B. Landfeldt, "Increasing road traffic throughput through dynamic traffic accident risk mitigation," *J. Transp. Technol.* vol. 5, no. 4, p. 223, 2015.

[30] E. Fitzgerald and B. Landfeldt, "A system for coupled road traffic utility maximisation and risk management using VANET," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 1880–1887.

[31] R. Ahmed, "Fuzzy risk-based decision method for vehicular ad hoc networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 9, pp. 54–62, 2016.

[32] H. Hasrouny, C. Bassil, A. E. Samhat, and A. Laouiti, "Security risk analysis of a trust model for secure group leader-based communication in VANET," in *Proc. Veh. Ad-Hoc Netw. Smart Cities*, 2017, pp. 71–83.
[33] J. Zhang, L. Huang, H. Xu, M. Xiao, and W. Guo, "An incremental BP neural network based spurious message filter for VANET," in *Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery*, Oct. 2012, pp. 360–367.
[34] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, "T2FA: Transparent two-factor authentication," *IEEE Access*, vol. 6, pp. 32677–32686, 2018.
[35] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-based distributed software-defined vehicular networks: A dueling deep Q-learning approach," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1086–1100, Dec. 2019.
[36] G. Stoneburner, A. Y. Goguen, and A. Feringa, "Risk management guide for information technology systems," NIST, Gaithersburg, MD, USA, Tech. Rep., 2002.
[37] K. Xiao, K. Liu, J. Wang, Y. Yang, L. Feng, J. Cao, and V. Lee, "A fog computing paradigm for efficient information services in VANET," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–7.
[38] G. Parmigiani and L. Inoue, *Decision Theory-Principles and Approaches*. Hoboken, NJ, USA: Wiley, 2009.
[39] R. Sugumar, A. Rengarajan, and C. Jayakumar, "Trust based authentication technique for cluster based vehicular ad hoc networks (VANET)," *Wireless Netw.*, vol. 24, no. 2, pp. 373–382, Feb. 2018.
[40] R. Azizi and G. Oz, "Performance evaluation of data dissemination in real-world wireless ad hoc networks," in *Proc. Int. Conf. Commun. Inf. Technol. (ICCIT)*, Mar. 2011, pp. 176–179.
[41] R. J. Atwa, P. Flocchini, and A. Nayak, "Risk-based trust evaluation model for VANETs," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Oct. 2020, pp. 1–6.

RASHA JAMAL ATWA (Graduate Student Member, IEEE) received the B.S. degree in computer science and engineering from Taibah University, Saudi Arabia, in 2009, and the M.S. degree in system science from the University of Ottawa, Canada, in 2016, where she is currently pursuing the Ph.D. degree in computer science with the School of Electrical Engineering and Computer Science. She is also a Staff Member of King Abdulaziz University and then, she moved to the University of Jeddah. Her research interests include trust evaluation in VANETs, smart cities, e-learning, and e-business. She awarded a full scholarship to pursue the graduate studies from her government.

PAOLA FLOCCHINI received the Ph.D. degree from the University of Milan, Italy. She is currently a Full Professor and the University Research Chair of distributed computing with the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. Her main research interests include theoretical computer science, specifically distributed computing with special focus on mobility (moving and computing) and on dynamicity (time-varying graphs). She is also interested in fundamental computational and algorithmic issues that arise among autonomous mobile computational entities, in the design of algorithmic solutions in the context of dynamic networks, and in sense of direction and other structural information. In 2019, she was awarded the Prize for Innovation in Distributed Computing.

AMIYA NAYAK (Senior Member, IEEE) received the B.Math. degree in computer science and combinatorics and optimization from the University of Waterloo, Canada, in 1981, and the Ph.D. degree in systems and computer engineering from Carleton University, Canada, in 1991.

He is currently a Full Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. His research interests include mobile computing, wireless sensor networks, and the Internet of Things. He is also an Associate Editor of IEEE Internet of Things Journal, IEEE Transactions on Vehicular Technology, IEEE Open Journal of the Computer Society, *Journal of Sensor and Actuator Networks*, *Future Internet*, and *International Journal of Distributed Sensor Networks*. He has served on the Editorial Board of several journals, including IEEE Transactions on Parallel and Distributed Systems, *International Journal of Parallel, Emergent and Distributed Systems*, and *EURASIP Journal on Wireless Communications and Networking*.