Towards Smart Trust Evaluation in VANETs

by

Rasha Atwah

Thesis submitted to the University of Ottawa

in partial Fulfillment of the requirements for the degree of

Doctor of Philosophy in

Computer Science

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

uOttawa

Abstract

With the dramatic growth of vehicles around the world, Vehicular Ad-hoc Networks (VANETs) have been proposed as a solution to advance road safety, improve transportation efficiency, and satisfy road users. In the VANET environment, vehicles communicate with each other and with road infrastructure in an ad-hoc manner. This communication may be safety-related or non-safety-related and may often include vehicle information (e.g., location, direction, speed, and control), road conditions, and events. A key component in assessing the veracity of the information is the trustworthiness of the information source. Thus, trust evaluation is one of the main requirements of VANET design. In this work, we investigate performance improvements in the trust evaluation framework of VANETs.

First, we propose a risk-based trust evaluation model (RTEAM) to estimate the risk of taking action or refraining from action regarding a reported event (in case of receiving conflicting messages about the event's existence). Some trust metrics such as direct trust, hop-based trust values, proximity to the event, and consequences of acting on a wrong decision are used to estimate the risk of the vehicle's actions. Vehicles make individual decisions by seeking the action with the lowest risk.

Second, we propose a fog-based reputation evaluation model (FREM) to support trust management framework. We promote fog computing as a new paradigm since it can provide several services to users in the edge layer. In our work, Fog supports the decision-making process in the reputation evaluation framework. Fog nodes play a key role in collecting vehicles' reputation records and cooperating with the roadside units (RSUs) to update these records. We propose the use of Digital Trustworthiness Cards (DTC), where the latest reputation evaluation of a vehicle automatically appears on its card. The benefits of the DTC are twofold: 1) the communication load on vehicles is reduced, and 2) historical trust records are established for each vehicle. We also take advantage of fog's familiarity and greater knowledge of the vehicles that frequently visit its zones; with more intimate knowledge, fog can smartly employ vehicles to perform specific tasks based on their experiences. Further, we implement a strategy for establishing trust based on specific task categories. This permits a nuanced evaluation of the vehicle best suited for the task at hand and has the further benefit of preventing malicious vehicles from being naively trusted based on successful completion of unimportant or non-safety-related tasks.

Finally, we expand the role of the fog in the decision-making process when vehicles need to ensure the existence of serious events. We propose a fog-based event validation model (FEVM) to validate the event's existence through cooperation between vehicles and fog nodes. The vehicles are used as mobile fog nodes, which compute their confidence in events based on the available information. Fog nodes then validate the event after combining vehicles' confidence values by applying the Extended Dempster-Shafer (EDS) theory of evidence. To test our proposed models, we conduct many experiments to investigate their performance and compare them with other existing models.

Acknowledgements

First and foremost, I would like to thank God for giving me good health and patience to persevere during my studies. This thesis would not have been possible without his care and guidance.

I must also express my profound gratitude to my beloved parents for their faith in me, their inspiration, and their understanding and support in every aspect throughout my graduate studies. I could not have reached this point without them. They are my undeterred source of strength and the reason I have made it so far in life. I wish to also thank my loving and supportive brother, my journey partner, Abdelrahman, for providing me with generous assistance, unfailing support and continuous encouragement through the process of pursuing my studies. Special thanks to my lovely, sweet daughter, Ajwan, for her endless love, support, and patience. Ajwan is more than a daughter to me. She is my little mother, best friend, and a gift that I always thank God for. My little man, Amjad, thank you for your smile that shines my life.

My little cutie one, Albaraa, you came to this world to be my super beautiful gift. You came in one of the most challenging times of my life. I had to submit my thesis after only one week from your birthday. I cannot believe that I have you after more than eleven years. I thank Allah for having you. May Allah bless you and your siblings. I am lucky to have you Jojo , Jody, and Bebo. My lovely husband Abdallah, thanks for your love, care, and being always beside me. May Allah bless our love and family.

I would like to convey my heartfelt gratitude and special appreciation to my supervisor, Prof. Paola, for her guidance, encouragement and unbreakable patience. I owe my deepest gratitude to my co-supervisor, Prof. Amiya, for providing me with an excellent learning and research experience. I appreciate his valuable assistance and comments during the progress of this work and for gently steering me in the right direction whenever he thought I needed it. Finally, I am forever grateful to the College of Computer Science and Engineering at Jeddah University for supporting me in pursuing my studies.

Dedication

This thesis is dedicated to Baba's memory, King Abdallah ben Abdelaziz, his endless love, care and support have sustained his people and country.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3G	Third-generation cellular networks
4G	Fourth-generation cellular networks
5G	Fifth-generation cellular networks
BS	Base Station
BTM	Beacon-based Trust Management system
CA	Certificate Authority
CAMs	Cooperative Awareness Messages
CFC	Coordinator Fog Computing
CH	Cluster Head
CTMs	Combined Trust Models
DCTMs	Data-Centric Trust Models
DSE	Dempster-Shafer Theory of Evidence
DSRC	Dedicated Short Range Communication
DTC	Digital Trustworthiness Card
ECTMs	Entity-Centric Trust Models
EDS	Extended Dempster-Shafer Theory
EO	Event Observer
EP	Event Participant
EP	Exploring Path
ER	Event Reporter
FEVM	Fog-based Event Validation Model
FLS	Fuzzy Logic System
FREM	Fog-based Reputation Evaluation Model
GPS	Global Positioning System
GV	Governmental Vehicle

HR	High Role vehicle
HTM	Hop-based Trust Model
ID	Identification Number
IoT	Internet of Things
ITS	Intelligent Transportation System
LoS	Line of Sight
LR	Low Role vehicle
MANET	Mobile Ad hoc Network
MCR	Multiple Conflicting Reports
MTM	Multi-faceted Trust Model
MUR	Multiple Unanimous Reports
NGV	Non-Governmental Vehicle
NIST	National Institute of Standards and Technology
NLoS	Non-Line of Sight
PKI	Public Key Infrastructure
PP	Preference Path
REM	Risk Estimation Module
RL	Risk Level
RSUs	Road-Side Units
RTEAM	Risk-based Trust Evaluation Advanced Model
RVM	Report Validation Module
SCM	Security Check Module
SR	Single Report
STM	Simple Trust Model
TEM	Trust Evaluation Message
TER	Task-based Experience Reputation
TPR	True Positive Rate
TRIP	Trust and Reputation Infrastructure-based Proposal
TS	Trust Server
UND	Undefined Cases

V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANETs	Vehicular Ad-hoc Networks
VFC	Vehicular Fog Computing
VFL	Vehicular Fog Layer
VFNs	Vehicular Fog Networks
VNL	Vehicular Network Layer

Chapter 1:
Introduction

1.1 Background

In 2016, there were about 1.35 million traffic fatalities worldwide, with the majority of victims being between 5 and 29 years of age [1]. At that time, the estimated number of vehicles in the world was around 1.32 billion vehicles counting personal cars, trucks, and buses. The number of registered vehicles is expected to reach 2.8 billion worldwide by 2036 [2]; subsequently, the number of accidents is also expected to increase considerably. Vehicular Ad-hoc Networks (VANETs) have been proposed as a solution to strengthen road safety, improve transportation efficiency, and enhance user comfort [21]. VANETs will also serve as a key component in the building of smart cities, see Figure 1.1.

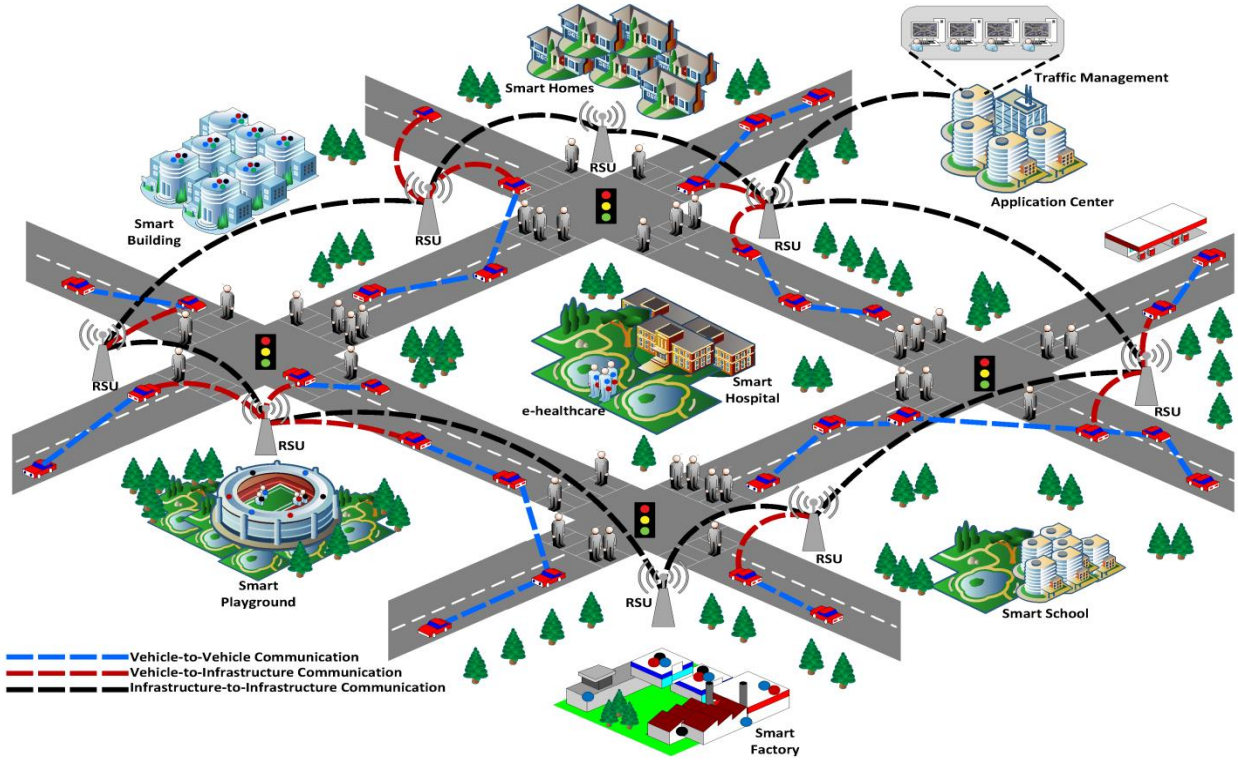


Figure 1.1: Illustration of VANETs in smart cities [30].

In the VANET environment, vehicles[1] communicate with each other in an ad-hoc manner (V2V) and with the road infrastructure (V2I), such as Road-Side Units (RSUs), through Dedicated Short-Range Communication (DSRC) radio [3]. This technology enables vehicles to exchange messages[2] in support of safety (i.e., accident warnings) and non-safety (i.e., entertainment) applications through periodically Cooperative Awareness Messages (CAMs) (i.e., beacon) [3]. Messages exchanged between neighbouring vehicles and infrastructure in the VANET may contain information about the vehicle, such as its location, direction, speed, and control information [5]. Consequently, security, privacy, and trust are among the main requirements of designing VANETs [6].

Reliance on social trust information to main safety and security in VANETs is an emerging trend because it overcomes issues in cryptography-based solutions that reduce network efficiency [8]. Trust dictates the level to which vehicles accept or rely on other vehicle messages and is held as the key element supporting security in vehicular networks [9]. In other words, vehicles must evaluate and establish trust among each other during communication, and this is called trust management. So, for any received message (i.e., safety or non-safety message), the receiving vehicle computes the trustworthiness of the sender and/or the received information. The receiver then decides to accept or reject the received message and take appropriate action regarding the information. Propagation of false/inaccurate information through the network may lead to minor or major issues on the road and affect the overall traffic safety and network efficiency. Therefore, reputation and trust establishment/evaluation in VANETs have also been studied extensively since VANET applications may use social trust in trust management [11]. Several solutions have been proposed to overcome the VANETs security threats [12].

Researchers have proposed various trust-based and reputation-based models to overcome VANET challenges, such as the short communication period (due to high mobility) and the random changes of the network topology. These proposed models differ in their architectures and trust metrics. However, to the best of our knowledge, there is no existing work so far that evaluates trust in combination with the consequences of wrong decisions to assess the risk associated with decision-making. Moreover, most existing models rely on nodes in the VANET to manage trust individually, which burdens the node's resources. On the other side, some researchers have incorporated RSUs to reduce the nodes' workloads and enrich data services in VANETs [13,14]. However, due to the short communication range and the sparse distribution of RSUs, these solutions suffer from intermittent connections [15]. Thus, few studies have incorporated the concept of fog computing in VANETs.

In this thesis, we explore trust evaluation models. We propose a novel risk-based trust evaluation advanced model and compare it with existing, purely trust-based models through simulation experiments. We introduce the concept of Task-based

[1]In this thesis, we used the terms "vehicle", "entity", "traveller", "user" and "node" interchangeably.
[2]We used the terms "information", "message", and "report" interchangeably.

Experience Reputation (TER) to support smart employment in VANETs and establish a smart reputation evaluation system that determines the node's competency for performing a specific task. We also propose a novel Fog-based Event Validation Model (FEVM) that supports the nodes' decision-making to validate a serious event on the road. In the remainder of this chapter, we present the motivation behind our research focus in Section 1.2, followed by our research problems in Section 1.3. We present our main contributions and the organization of the thesis in Section 1.4 and Section 1.5, respectively.

1.2 Motivations

Over the upcoming decades, the expected rapid growth in the number of registered vehicles around the world [2] raises tremendous challenges for the traffic management authorities to effectively manage this heavy traffic load [19]. The new generation of vehicles will be equipped with onboard wireless devices (e.g., Bluetooth), internal and external sensors (e.g., radar), and positioning system devices (e.g., GPS)[3]to help the travellers become more aware of the road conditions and events around them.

The concept of VANET (i.e., vehicles and infrastructure relying on wireless communication and sensors network [18]) is offered to allow these vehicles to effectively communicate and cooperate by sharing and exchanging traffic information throughout the network (i.e., via safety and non-safety messages). This communication produces vast traffic data (traffic smoothness, events on the road, etc.) used to support various safety applications (e.g., event detection) and entertainment applications (e.g., chatting) in the VANET. The entertainment applications target the social component of VANET (i.e., a social network containing hundreds of users gathering in the same geographical area) to enhance the enjoyment of travellers while travelling on the roads. On the other hand, safety applications seek to preserve life by reducing traffic accidents by 30-70% [18]. Thus, the intelligent transportation system (ITS) takes advantage of the available data in VANETs to improve its efficiency, offer reliable options for travellers, and provide enhanced security[25]. Generally speaking, VANETs support the ITS to improve its performance[24].

However, high mobility and low connectivity (i.e., short communication time) stand as one of the main challenges in VANETs and make the VANET environment a fertile soil to spread false information that may threaten the safety of its users. Thus, the importance of trust evaluation between vehicles is evident and has attracted considerable interest lately. Before believing received information, the sender of this information and the received data should be trusted by the receiver.

More recently, the idea of using fog computing has gained much research interest in works related to VANETs. This interest has surged because fog computing can support safety applications such as detecting malicious nodes and false messages.

[3] http://www.intechopen.com

However, until now, fog computing has not taken into consideration the trust evaluation framework in VANET. In other words, fog with its capabilities (e.g., data storage and computing) has not yet given a significant role in trust evaluation in VANETs.

In summary, this work is motivated by the VANET's role as a key component of smart cities in the near future [25] and, furthermore, the potential for VANETs to improve traffic safety, decrease traffic accidents and save lives through effective evaluation of trust and smart employment of vehicles throughout the network.

1.3 Research Problems

Devising and implementing strategies for how vehicles can trust their neighbouring vehicles and associated messages (i.e., Trust Management) [26] is a challenging task due to the ad-hoc nature of VANET (i.e., high mobility and low connectivity). The main goal of any trust evaluation model is ensuring the safety of the connected vehicles (i.e., the sender and the receiver) by assessing the trust level of the transmitting vehicle and/or its transmitted information before accepting or believing any reports from that vehicle. Various models have been proposed for this purpose (i.e., trust evaluation and management), which differ in their approaches and architectures. However, the following points highlight some of their limitations:

- Existing entity-based trust models evaluate the trustworthiness of the entity itself (i.e., the vehicle), whereas data-based trust models evaluate the trustworthiness of the data reported by the vehicle. In entity-based trust models, the issue of low connectivity may leave vehicles with insufficient time to acquire information about a neighbouring vehicle required for evaluating its trust. On the other hand, the main limitation of data-based trust models is that the models cannot work if the required information is incomplete or redundant [3]. Hybrid trust models combine both entity-based and data-based models and inherit the disadvantages of the entity and data-based trust models.
- Existing trust evaluation models decide to follow or ignore incoming messages based on the outcome of the trust evaluation of the sending vehicle, data, or both. However, the risk of the associated action regarding such decisions (accepting or rejecting the existence of an event) is not considered.
- Malicious or selfish nodes in the network may spread false messages that affect decisions by other nodes (the decision may be contrary to their observations), leading to cascading of wrong decisions and/or oversampling some decisions throughout the VANET. In existing models, the cascading and the oversampling problems are not addressed [19].
- Due to the high mobility and the rapid topological changes in VANETs, vehicles have short communication times to exchange messages. According to [18], the phenomenon of low connectivity is particularly common in areas with low traffic density (e.g., highways) and areas with high traffic density and short communication distance (e.g., due

to obstacles such as large buildings). Thus, the existing models are prone to issues of limited connectivity, which may impede the effectiveness of trust management and other processes in the VANET environment.

- In VANETs, nodes must establish trust neighbouring nodes in one of the following ways: 1) Assuming arbitrary initial trust (to avoid cold start), or 2) Posing a query to which the true answer is already known (i.e., direct experience). However, the first method cannot reflect the actual trust level of the node [27]. Also, in the second method, the malicious node can correctly answer the test question to trick its neighbour (i.e., the attacker pretends it is a legitimate node). Additionally, method two is at odds with the high mobility nature of the VANET, where vehicles may have only one chance to communicate with their neighbours.
- Most existing trust evaluation models focus on real-time experience; however, after the evaluation is complete, the historical records vanish from the vehicle's memory, precluding the future use of these rich records. Hence, from the user's perspective, communicating with other neighbours to evaluate the trust through direct or indirect experience for every session is time consuming and wasteful of resources.
- Experience-based trust evaluation models use the term "experience" to broadly define any communication among the vehicles regardless of the type of experience or its effect on the network. Therefore, malicious nodes can gain trust credits with good performance/attitude in basic or non-safety-related communications leading to trust in malicious nodes. Indeed, using experience-based trust as a general metric without considering the experience type and its effect on the network allows malicious nodes to foil VANET objectives by posing as trustworthy nodes and then corrupting safety-related reports/functions at opportune times.
- Event warning messages may spread at any time in the network. Two possible scenarios may unfold: 1) All messages agree with the event or deny the event, or 2) Some messages agree with the event, and some deny the event (i.e., conflicting messages). The first scenario is generally used in most trust-based event detection models, where the focus is on computing the trust level for the message senders, the context of the messages, or both. However, the second scenario, where conflicting messages (regarding an event) are available, is not appropriately covered in most existing work, although it is a realistic scenario.
- Fog computing holds the promise of significant potential to edge users in reducing response time and required storage capacity; however, at the time of finalizing this thesis, the trust management frameworks in VANET generally do not fully benefit from fog capabilities.

This work, therefore, addresses the following research questions: Can the "Risk" metric drive "Trust"/ "Decision making" in the trust evaluation framework? Can a "Purely trust-based model" or a "Risk-based model" enhance the outcomes of the decision-making process? What services can fog provide to support the reputation evaluation system in VANET? Can long-

term trust (i.e., reputation) provide an effective alternative to the arbitrary initial trust value? Does the aggregated or accumulated reputation value reflect the actual trust level of the vehicle? Can fog support the event validation process? Can vehicles be utilized as mobile fog nodes to support the event validation process?

1.4 Contributions and Scholarly Achievements

This section summarizes our proposed models and identifies how our work contributes to addressing the research problems.

1.4.1 RTEAM: Risk-based Trust Evaluation Advanced Model

Our approach:

We divide our work into two main approaches, each with its own contributions. In the first approach, we express a method to assess the risks of accepting or rejecting information regarding a purported event. The risk estimation is based mainly on how close the vehicle is to the purported event. We designed our model based on a realistic scenario in which conflicting reports are propagated throughout the network. In addition to driving decision-making with risk assessment, the model also rectifies an issue commonly plaguing purely trust-based models where an event's state sometimes becomes indeterminate based on the available trust metrics. Additionally, we add a new trust metric (hop-based trust) to improve the accuracy of the model and avoid the oversampling issue (i.e., a vehicle weighing information from other vehicles without recognizing that the vehicles have influenced each other). Experiment results demonstrate that integrating risk estimation with trust evaluation provides reliable detection of events combined with lower risk exposure.

Contributions:

- Our proposed model (RTEAM) detects the event state according to the associated risk of the actions associated with believing or disbelieving the occurrence of the reported event.
- RTEAM uses trust metrics to drive the likelihood of the estimated risk of an action where the impact of that action is computed according to the proximity from the reported event and the number of affected neighbours.
- RTEAM is a lightweight model that reduces processing time, saves resources, and consumes energy by following two phases filtering scheme. Phase one filters all invalid reports. Then, phase two, where RTEAM ensures the sender's authentication and trust level before accepting the report.
- RTEAM can determine the event state all the time, unlike the simple voting (the majority opinion is followed) and multi-faceted trust model, where the event state cannot be detected if the number of event believers and deniers are equal, or the majority consensus is not reached, respectively.

1.4.2 FREM: Fog-based Reputation Evaluation Model

Our approach:

Our second approach uses fog computing to support the VANET environment and benefits vehicle-based applications in terms of response time, communication, and storage[29]. In our model, fog nodes are considered the main points that aggregate the nodes' reputation to support the trust evaluation framework in VANETs by reducing the workload on the nodes (i.e., no need for real-time experience). Also, we shed light on a function that local fog nodes can perform when it comes to the local people (i.e., vehicles). Fog nodes identify well-known travellers of their zones (i.e., people who live or work in the area) so they can potentially employ people based on this knowledge.

Contributions:

- Our proposed model reduces the workload on the vehicle by allowing them to exchange their reputation values. Moreover, it reduces the transmission message overhead.
- We address the problem of combining all evaluations of different experiences into one aggregated or accumulated trust value. Also, we propose a simple solution to solve this problem.
- The concept of Task-based Experience Reputation (TER) is proposed to prevent selfish and malicious nodes from bolstering their reputations while seeking harmful goals.
- By using TER, the infrastructures and the vehicles can depend on the most competent node, not simply the most trusted.
- Fog can do more to support the trust evaluation framework due to its capabilities and position on the network.
- Historical trust records are rich sources that should be leveraged to provide more accurate and reliable evaluations for the nodes.

1.4.3 FEVM: Fog-based Event Validation Model

Our approach:

The FEVM is our third proposed model that extends the functions of fog nodes by appointing decision-making authority to fog nodes in the VANET. Fog plays the main role in event validation and uses vehicles as mobile nodes to identify and form confidence values for events. Most vehicles participate in the event validation by forming confidence levels regarding the event, then fog aggregates the evidence from the participants to decide on the truth of the event. Vehicles compute their confidence levels based on metrics such as role-based trust, and their relationship to the event (i.e., are involved in the event, observed the

event, or received reports about the event) and based on the reports they received. Finally, fog collects all event confidence levels from the vehicles by applying Extended Dempster Shafer (EDS) to estimate the event state and take appropriate action.

Contributions:

- Utilizing fog unifies the actions of the vehicles and prevents disordered vehicle behaviour.
- FEVM can handle the cases where there is a single report, multiple similar reports, or multiple conflict reports, which guarantees that a decision can always be made regarding an event (i.e., all scenarios are covered).
- Fog can work independently (without the need to hear from its vehicles) if other fog nodes or IoT devices confirmed the event's occurrence.
- Not all vehicles must participate in the event validation process. In our model, fog can validate events and make decisions without involving all vehicles in the process.
- Fog can recognize different levels of trust, confidence, etc., in received reports when combining their evidence by assigning importance factors to the reports accordingly.
- The decision threshold can be tweaked for the promptness of decision-making and optimized for the conditions of the VANET (e.g., shorter threshold when there is a high prevalence of untrustworthy vehicles).
- Fog can continue to re-assess its decisions by collecting further evidence.

The contributions investigated in this thesis are summarized as follows: 1) decision-making evaluations in the VANET can be determined according to the associated risk of prescribed actions, 2) smart reputation systems can be built based on the Task-based Experience Reputation, putting us on the path towards smart employment in VANET, 3) fog can play a vital role in the trust evaluation framework, and 4) fog can validate serious events on the road.

In the process of completing this work, the following publications have been produced.

Conference Proceedings:

- Rasha Atwa, Paola Flocchini, and Amiya Nayak, "Risk-based Trust Evaluation Model for VANETs." In Proceedings of IEEE International Symposium on Networks, Computers, and Communications, 2020. (Chapter 3)
- Rasha Atwa, Paola Flocchini, and Amiya Nayak, "Towards Smart Trust Management of VANETs." In Proceedings of the 33rd IEEE Canadian Conference on Electrical and Computer Engineering, 2020. (Chapter 4)
- Rasha Atwa, Paola Flocchini, and Amiya Nayak, "A Fog-based Reputation Evaluation Model for VANETs" In Proceedings of IEEE International Symposium on Networks, Computers, and Communications, 2021. (Chapter 4)

Journal Publications:

- Rasha Atwa, Paola Flocchini and Amiya Nayak, "RTEAM: Risk-Based Trust Evaluation Advanced Model for VANETs, IEEE Access 9: 117772-117783 (2021). (Chapter 3)
- Rasha Atwa, Paola Flocchini, and Amiya Nayak, "FEVM: Fog-based Event Validation Model" (Chapter 5) – (in preparation for submission).

1.5 Roadmap

The road map of the rest of this thesis is outlined below:

- Chapter 2**, Background and Related Works,** reviews the literature on topics related to our research problems. We provide an overview of trust evaluation in VANETs, survey the existing works according to the categories of trust evaluation models in VANET, and summarize the limitations of each approach. Finally, we survey some works that use fog computing to support the trust evaluation framework in VANETs.
- Chapter 3**, RTEAM: Risk-based Trust Evaluation Advanced Model,** presents a novel trust evaluation model for detecting the event state (i.e., "Exist" or "Does not exist") in VANET based on the estimated risk of the action taken in the face of uncertainty. We present extensive experiments to show that RTEAM leads to better system performance, such as lower overall risk.
- Chapter 4, **FREM: Fog-based Reputation Evaluation Model,** addresses the problem of the traditional trust evaluation models in VANETs, where the attacker can exploit the existing evaluation models to gain credits and raise its trust level. The concept of task-based experience reputation is proposed to prevent selfish and malicious nodes from raising their trust levels by categorizing their tasks instead of putting all the values in the same boat. The chapter also provides a state-of-art of fog-based reputation evaluation model supported by fog computing.
- Chapter 5**, FEVM: Fog-based Event Validation Model,** presents a novel reputation-based event validation model for detecting the event state (i.e., "Exist" or "Does not exist") in the VANET based on the collected pieces of evidence from the vehicles. We present extensive experiments to show that FEVM leads to better system performance in terms of TPR and UND compared to some other models such as RTEAM.
- Chapter 6**, Conclusion and Future work,** contains a summary of our thesis by concluding all the contributions and limitations and presenting possible future work.

Chapter 2:
Background and Related Works

2.1 Trust and Reputation in VANETs

According to social science, humans have a natural disposition to trust and judge trustworthiness due to their neurobiological structure and brain activity. In a social context, trust refers to the situation where we have a trustor, trustee, and a situation regarding the future. The trustor will be willing to rely on the trustee's actions to make a decision regarding a situation if the trust is great enough [31,32]. In other words, "Trust" reflects the level of confidence in or dependency on a person or a thing to do a special task. In the computer science context, trust has been borrowed from social science literature, and until now, has no standard definition [33]. Nevertheless, the concept of trust is utilized to improve security in ad-hoc networks [34], and some researchers have mentioned trust as a key element of security in ad-hoc networks such as [9].

Reputation is another concept appearing in existing works regarding VANET security. The management of trust and reputation in combination has been proposed as a novel and original way to address some network security threats [38]. However, trust is not technically a form of reputation; trust reflects the level of confidence, reputation reflects historical interactions. Though the concepts are different, they are related. For example, entity *A* may trust or have confidence in entity *B* because entity *B* may have a good reputation. In short, the relationship between trust and reputation could be interdependent to the extent that trust may be established based on reputation.

In Table 2.1, we show some definitions of trust from existing works. We have highlighted the keywords from the existing definitions to compose a new definition of trust that is consonant with the main aspects of the trust concept in VANETs and our work development. In Chapter 3, RTEAM makes its decision to believe or disbelieve the event's occurrence based on the vehicle's experience-based, role-based, and hop-based trust. In Chapter 4, the fog-based proposed model uses the vehicle's Task-based Experience Reputation (TER) as a trust indicator. Whereas in Chapter 5, FEVM makes its decision to confirm the event's occurrence based on the vehicle's confidence level and its event confidence level (i.e., available evidence). Based on the definitions in Table 2.1, we define trust as follows:

Trust: the level of confidence in or willingness to depend on an entity based on experience, available evidence, reputation, and/or recommendations.

Table 2.1: Existing definitions of trust in VANETs.

Study	Trust Definition
[10]	Trust describes the level to which an entity accepts the dependence on another one.
[33]	The belief that an entity has about other entities, from past experiences, on knowledge about the entity's nature, and/or on recommendations from trusted entities.
[36]	Trust can be described as expectation and belief about future behavior, based on experiences and evidences collected in the past, either direct or indirect.
[37]	Trust is a relation among entities that is established based on the observations of historical interactions.

2.2 Trust Evaluation Models

In the last decade, the concept of trust management has become a hot topic due to its importance in securing the application's integrity and reliability [39]. Several trust models have then been proposed to enhance VANET security and are grouped into one of three categories based on their trust evaluation mechanisms into three categories: 1) entity-centric, 2) data-centric, and 3) combined trust models [8]. In the following subsections, we present the three categories of trust models in VANETs and explore relevant research works for each category.

2.2.1 Entity-Centric Trust Models

Entity-centric trust models (ECTMs) evaluate the trustworthiness of the vehicles [40], where the trustworthiness of information is estimated based on the trustworthiness of the sender of the information [48]. The trustworthiness of the sender can be estimated based on many parameters such as the node's experience with its neighbours (direct and/or indirect), role (i.e., governmental, public transportation, or ordinary vehicle), location, speed, direction, reputation, etc.

In cluster-based approaches, the elected cluster head (CH) leverages for trust computing and/or aggregation as done in [41, 42]. However, in non-cluster approaches, the node itself is responsible for establishing the trust metric of the targeted neighbour, as done in [38, 43, 44].

In [43], Minhas et al. proposed a multi-faceted trust model that incorporates role-based, direct experience-based, priority, and majority-based trust to predict the event's state and make a real-time decision. In this strategy, neighbouring vehicles were 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 at hand (priority) before sending requests to selected neighbours. Once the node

receives all responses, a majority consensus is applied to estimate the event's state (i.e., event occurred or not). If the majority consensus exceeds a threshold, the node accepts the advice. Otherwise, it takes the opinion of the sender with the highest role/experience trust. Each node must apply Public Key Infrastructure (PKI) to get the role-based trust [8], which imposes an overhead on the nodes. Also, the main limitation of this approach is that it cannot detect the event's state if two vehicles with conflicting reports have the same role/experience-based trust. The nodes in Minhas et al.'s proposal were responsible for the trust evaluation; however, other existing research employs the static infrastructure (i.e., RSUs) to estimate the trust values, distinguish the malicious nodes from the trusted ones, and support the communication and trust management framework. The latter strategy was proposed because RSUs have a higher transmission range and larger storage capacity than vehicles; thus, it can see the big picture of the network. Therefore, relying on secure RSUs in trust evaluation or establishment processes can reduce the vehicles' communication overhead.

Marmol and Perez [38] proposed a trust and reputation infrastructure-based trust model (TRIP) where RSUs were responsible for the trust evaluation process. RSU estimates reputation scores based on experience trust (direct trust), recommendations from the node's neighbours, and recommendations from a central authority. However, the model lacked privacy and security measures [59]. Similar work in [44] employed RSUs to establish trust, oversee the nodes' behaviour, and share it with vehicles when requested. Furthermore, this model used direct and indirect trust, recommendation and reputation to estimate a vehicle's trust value. The major drawback of ECTMs is that vehicles may not have enough time to acquire the required information to evaluate the trust of a neighbouring vehicle.

2.2.2 Data-Centric Trust Models

DCTM gathers information from the network, which 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 assigns different weights to the nodes based on the number of hops from the event. In other words, nodes that are one hop from the event are weighted more heavily than nodes two hops or more from the event. The proposed model also overcomes the oversampling and cascading issue (i.e., some vehicles influence other vehicles' opinions).

In [46], Ding et al. proposed an event-based reputation model that recognizes different vehicle different roles (i.e., event reporter, event observer, and event participant). Then, based on the vehicle's role, the vehicle evaluates the trustworthiness of the received message. Moreover, RSUs are used to manage the long-term trust for vehicles that commonly use the same path. However, when the RSUs cannot provide a trust value for a vehicle (such as when a vehicle first uses the route), an event-centric mechanism is applied [47]. Other models use various pieces of information to evaluate the trustworthiness of the message, such as content similarity, content conflict, routing similarity [48], distance calculation and the vehicle's geolocation

[49], or location/time closeness and location/time verification [50]. The DCTMs cannot work if the required information is incomplete or redundant [3]. Also, latency and data sparsity are other limitations of DCTMs [3].

2.2.3 Combined Trust Models

Combined trust models (CTMs) evaluate the trust in the entity to competently compute the trustworthiness of the data [9]. Therefore, CTMs inherit the drawbacks and the benefits of ECTM and DCTM [3]. A multi-layer fuzzy-logic-based model proposed by Solyman et al. [51] estimates the direct experience and the sender's plausibility information (i.e., location and time). It then detects malicious nodes and tackles data uncertainty in the vehicular network in both Line of Sight (LoS) and Non-Line of Sight (NLoS) cases. The authors also employed fog nodes to ensure the accuracy level at a very late stage. However, employing fog at earlier stages of the process can conserve time and resources.

A beacon-based trust management system (BTM) proposed by Chen and Wei [37] aims to prevent spreading false messages via VANET. The entity trust (the message's sender) is constructed from a beacon message by finding the cosine similarity between the claimed and the estimated values of the vehicle's position, speed, and direction. Then, event-based trust is computed where a position and movement verification mechanism verifies the event's location and the vehicle's movement (the sender). Indirect event-based trust is computed in a manner that produces smaller trust values for nodes that are relaying the information (i.e., the sender is not the generator). Then, event reputation is computed, and the composite trust value is calculated. Finally, the Dempster-Shafer theory of evidence is used for combining the opinions. Solyman et al. [51] used direct experience trust to estimate the trustworthiness of the entity, whereas Chen and Wei [37] used a 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 approach. Clearly, based on [37] and [51], the sender's plausibility information is a useful and significant part of estimating trustworthiness.

2.3 Trust and Risk

According to [72], two contrary views can specify the relationship between the concepts of "Trust" and "Risk" as follows: 1) risk drives trust, or 2) trust drives risk. The first view holds when we have to rely on an entity (i.e., trust the entity) due to the low expected risk to deal with a situation. The second view holds when we expect that an entity will behave well (according to our knowledge or experience) in a situation. Thus, our interaction with this entity will not be risky or at least will not be very risky. Thus, the action resulting from the decision-making process is a combination of trust evaluation and risk estimation processes, as shown in Figure 2.1.

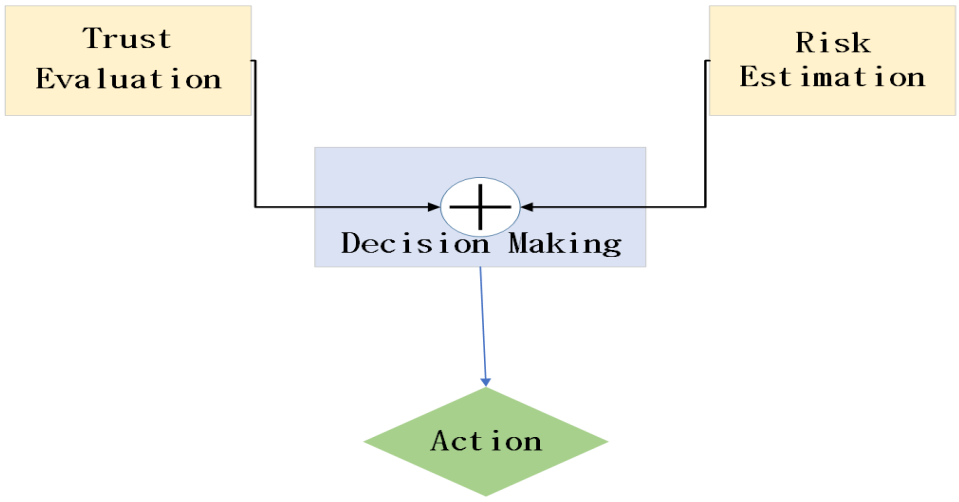


Figure 2.1: Relationship of trust evaluation, risk estimation, and decision-making process, and action.

Risk estimation in the VANET environment has been studied from different points of view: 1) Security perspective [52], and 2) Application perspective (i.e., The predicted risk of traffic accident) [55-57]. Dandan et al. in [52] 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 the victim's location information) based on the attack cost, technical difficulty, and probability of being 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 [53], where the security risk assessment framework is based on a conventional security analysis model and attack tree. The risk assessment is based on assets, threats, and vulnerabilities. The authors in [54] provided a context-based risk assessment sheet that can be used for VANETs, where threats were identified according to mobility.

From the application perspective, Emma and Bjorn in [55] proposed a traffic accident risk mitigation based on the neighbours' factors such as the driver, vehicle, and environmental conditions. Each vehicle makes its decision individually. Later, the authors modified their model in [56] by proposing one that takes the risk estimates of the surrounding neighbours into account. Similar work has been proposed in [57], where the risk was estimated based on the road traffic safety level instead. The risk was derived based on the sensitivity and type of application and then measured in a quantitative and qualitative manner by considering 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 using risk assessment has been applied in [8], where the authors used their context-based risk assessment sheet in [54]. Similar work has been done in [58], where the proposed security risk assessment was applied to their trust model. The

authors of [58] identified the possible threats in VANET, the risk level, and their countermeasures. For example, when the risk of replying to old messages is high, the use of a timestamp should be mandatory within the trust model. Thus, we can conclude from the works mentioned above that the risk level of security threats is estimated separately from trust evaluation. Moreover, the risk metric can be integrated with the trust management model to enhance the outputs. In this work, we mainly aim to take advantage of the trust metric to estimate the risk of taking action regarding an event.

2.4 Fog Computing in VANETs

One of the main objectives of designing smart cities is providing high-quality services for the users around the city. Such services from the end-user perspective mean achieving the user's goal with acceptable response time and in a professional manner. In the transportation segment, this cannot be done without efficient infrastructures and technologies designed with capacity for the expected explosive increase in the number of smart vehicles. Cloud computing and RSUs enable this vital role by providing easy access to network resources but the dramatic growth in end-user smart devices has increased the burden on the availability of storage, computing, and network resources [60].

To meet this ever-increasing demand in storage and computational capacity, the use of existing technologies such as third (3G) and fourth-generation cellular networks (4G) [61] up to (5G) and mobile cloud computing[62] was an appealing idea. However, these technologies are not sufficient [63] for the following reasons. From an application's perspective, cellular networks are not efficient because they may suffer from high data traffic charges and long services delays [15], while also being controlled by network operators[61]. From the user's perspective, mobile cloud computing is costly and time-consuming when the user must upload real-time information [64]. Besides, these technologies require high-quality internet connections with remote infrastructure[65]. This gave rise to turning to Vehicular Fog Networks (VFNs), which is a decentralized computing paradigm [60], as shown in Figure 2.2.

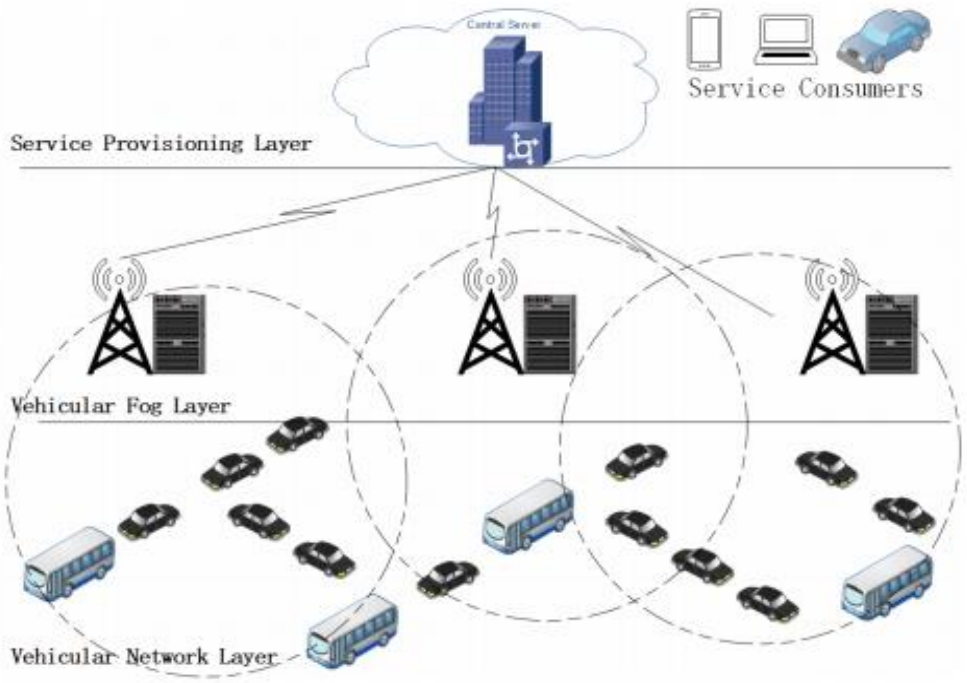


Figure 2.2: Vehicular Fog Networks (VFNs) architecture [45]**.**

Cisco [66] introduced a new computing term called *fog computing* that moves the functions of cloud computing (e.g., computing, storage) from the center to the edge of the network [67]. According to VFN structure, fog nodes (i.e., edge nodes [60]) are deployed at the edge of the network to effectively gather, organize, store, and process data in real-time [60]. From our literature review, we found that fog nodes are deployed in different positions in VANETs according to the required function. For example, in the Vehicular Fog Layer (VFL), fog nodes and RSUs work together as an intermediate layer between the cloud server and end-users. Also, they can communicate directly with the end-users in Vehicular Network Layer (VNL), as shown in Figure 2.2**.** Alternatively, some researchers conceived the idea of utilizing fog nodes as a separated layer from RSUs, such as [63, 68, 69], as shown in Figure 2.3.

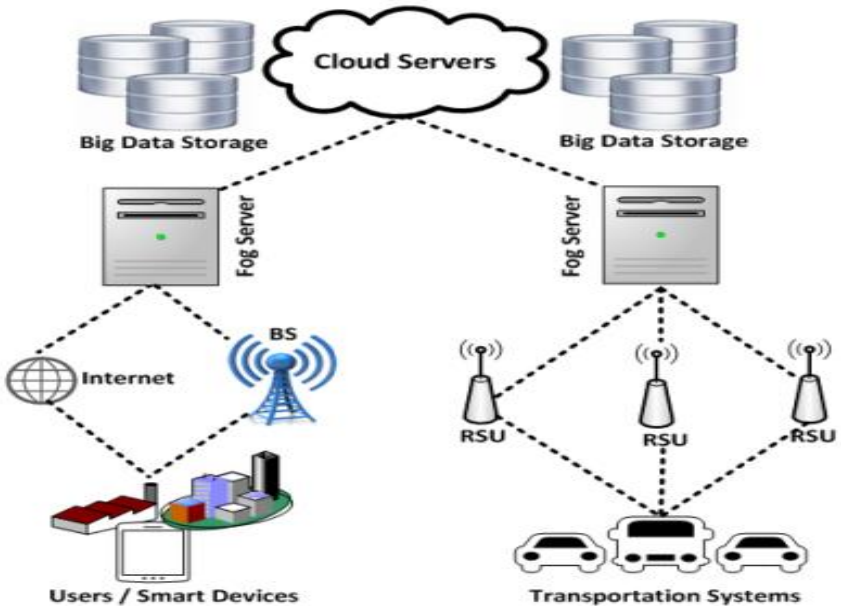


Figure 2.3: Fog server-based architecture for ITS using RSUs, Base Station (BS), and the Internet [69]**.**

Moreover, the existing literature focuses on benefitting VANETs with intelligent vehicles (with their built-in onboard sensors and smart devices) by deploying them as mobile fog nodes. Unlike VFNs in Figure 2.2, where fog nodes are utilized as fixed infrastructure, vehicles in Figure 2.4 are utilized as mobile infrastructure for communication and computation in what is called Vehicular Fog Computing (VFC) [63, 71]. Even though these vehicles are not moving (i.e., they are parked), they can support larger flow capacity[63] and maintain information regarding surrounding vehicles [71].

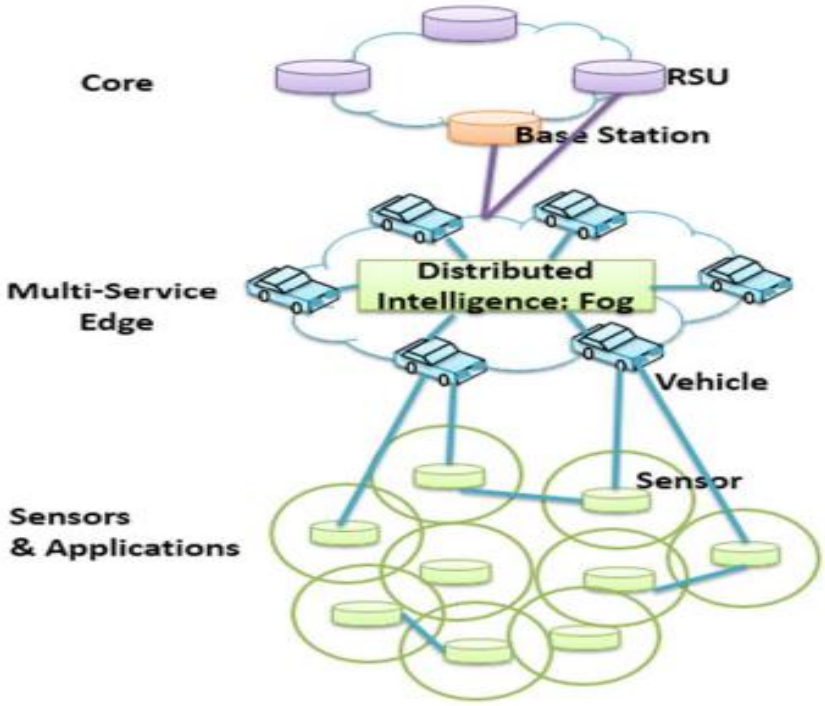


Figure 2.4: Vehicular Fog Computing (VFC) architecture [63]**.**

The question of what fog nodes can provide to VANETs is answered by authors of [70], where four services can be provided in what they called Coordinator Fog Computing (CFC) for the edge nodes in the network: 1) mobility support and control, 2) multi-source data acquisition, 3) distributed computation and storage, and 4) multipath data transmission.

The above literature shows that integrating fog computing in VANETs can enhance traffic applications such as parking and message dissemination in terms of latency and efficiency. However, few existing works focus on the role that fog can play in the trust evaluation framework of VANETs. Some researchers give fog a minor role to ensure the event's existence after evaluating trust, such as [51]. On the other hand, other researchers assign fog a major role in the trust evaluation framework by storing and analyzing data in VANET, such as [73] and detecting the malicious nodes such as [18]. Thus, there is room for expanding the role of fog in the trust evaluation framework, which is the focus of this work.

Chapter 3:

RTEAM: Risk-based Trust Evaluation Advanced Model

3.1 Introduction

Safety applications in VANETs (e.g., traffic violation, hazardous location notification, collision risk warnings, etc.) are mainly aimed at reducing accidents and saving lives [74]. Therefore, these applications are primarily grounded on cryptography, which is the first line for most network attacks [28]. However, cryptography cannot stand alone in front of all the security threats. Thus, trust between the nodes is an essential requirement to deal with security-related problems that cryptography cannot detect, such as fake messages and dishonest users [28].

As highlighted in the previous chapter, many works have been done in the area of trust evaluation in VANETs. The existing trust models evaluate trust are based on the available information about the report sender and/or the included information in the report. In the case of safety applications such as hazardous location notification, trust is evaluated, then the user (i.e., the driver) has to take the appropriate action such as changing its route or slowing down its speed. Most of the recent work considers the case where all nodes in the network report the same information regarding an event. For instance, all the vehicles report an accident occurrence and no single vehicle reports the opposite. In other words, the case of receiving conflicting reports regarding an event (i.e., "accident exists" and "accident does not exist") does not take into considerations in most of the existing models.

The concept of risk estimation is explored in VANET literature from the network perspective (i.e., the possible risk level for security threats) and the application perspective (i.e., the predicted risk of an accident). The idea of integrating risk level with trust evaluation framework has been suggested in [57] to enhance the performance of the trust management framework. Thus, based on this suggestion, we propose a risk-based trust evaluation advanced model called RTEAM, which identifies the occurrence of an event based on the risk of taking action regarding an event. RTEAM model is a risk-based trust model that uses the trustworthiness of the sender vehicle (i.e., entity-based) and the trustworthiness of the received information (i.e., data-based) to estimate the possible risk of believing or disbelieving the sender's report. Once a vehicle receives conflicting reports about an event, RTEAM checks the validity of the reports one by one (i.e., the report should be active and relevant to process). Then, RTEAM assesses the authentication of the sender and its trust level as the main security requirements before accepting

any information from that sender. Next, the risk of taking action (i.e., accepting or rejecting an event) is estimated, and RTEAM takes action with the associated lowest risk. Finally, the vehicle spreads its decision (i.e., event or no event) to its neighbouring vehicles. RTEAM is an example of the "*Risk drives Trust*" view where the user takes its decision and trusts the report based on the associated risk of the action of believing or disbelieving the report.

This chapter is organized as follows: the details of the proposed model are presented in the following section. We give an illustrated example to compare our work with others in Section 3.3, followed by performance evaluation in Section 3.4, discussion in Section 3.5, and chapter conclusion in Section 3.6.

3.2 Proposed Model

The proposed model (RTEAM) consists of three modules, as depicted in Figure 3.1:

1) *Report Validation Module* (RVM),

2) *Security Check Module* (SCM), and

3) *Risk Estimation Module* (REM).

The vehicle receives conflicting reports about an event from neighbouring vehicles. The report includes the event information (e.g., event stamp time, expiration time, etc.) and the sender's information (e.g., ID, direction, etc.).

First, RVM checks the report validity, then SCM checks the sender's authentication and the trust level of the sender. Both RVM and SCM are used for filtering invalid reports and unauthorized and/or distrusted senders, respectively. The report should be valid, and the report's sender should be authorized and trusted; otherwise, the report is discarded. Finally, REM computes the risk of taking an action based on computing the likelihood and impacts of each action of believing or disbelieving the received report (i.e., "Event" or "No Event"). 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:

1) The proximity of the vehicle (i.e., the report receiver) to the reported event and

2) The number of neighbouring vehicles that are affected by the vehicle's decision.

The vehicle then computes the risk of both actions corresponding to believing or disbelieving the event and makes a decision (i.e., taking the action with the lowest estimated risk). Each vehicle spreads its opinion (i.e., "Event" or "No Event") regarding an event to its neighbours that may affect by its report. We delineate the three modules RVM, SCM, and REM in detail in subsections 3.2.2, 3.2.3, and 3.2.4, respectively.

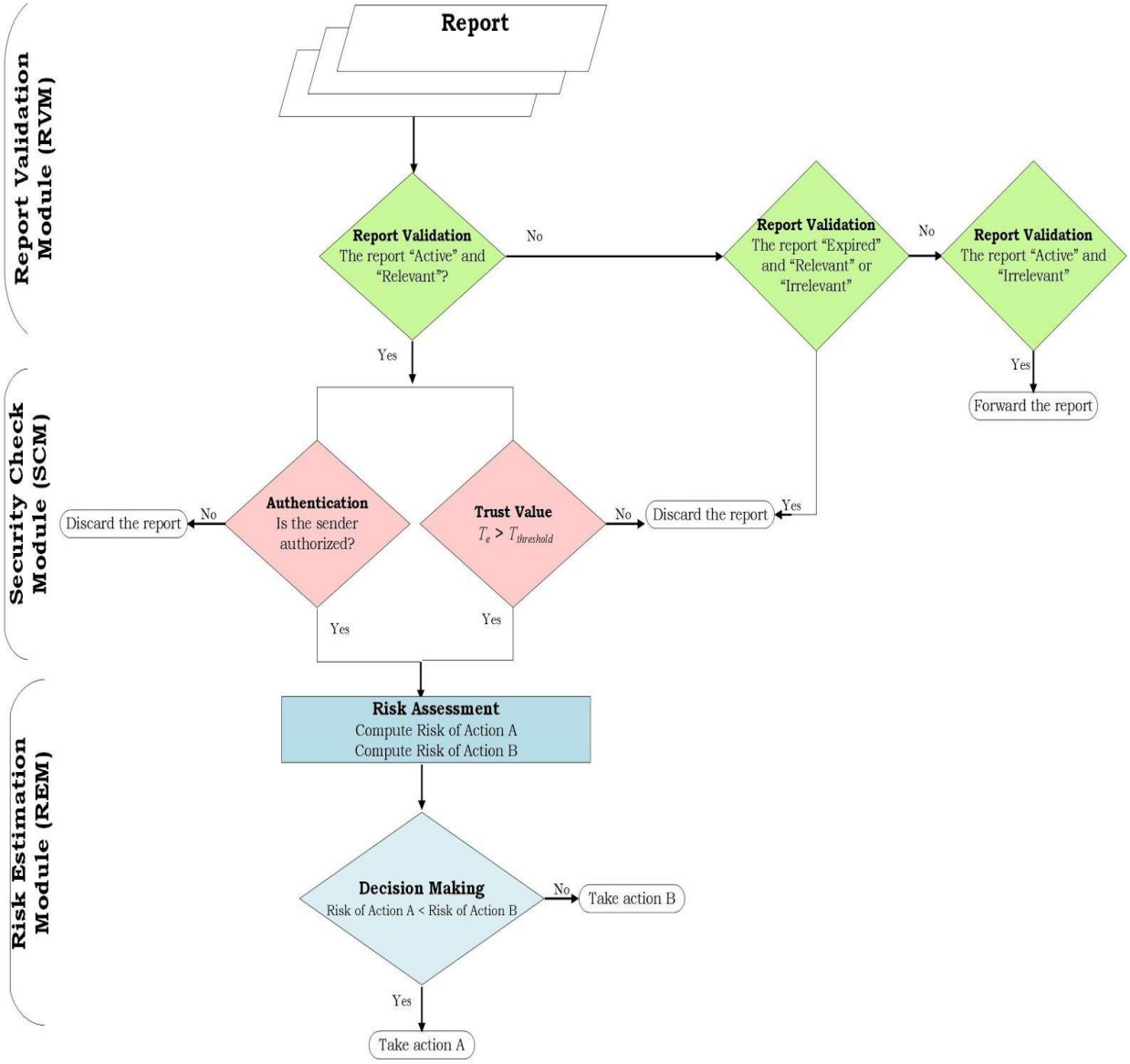


Figure 3.1: The framework of RTEAM.

3.2.1 Network Model

First, we give the notations of our network. Let v_i be a vehicle and v_j be any neighbour vehicle of v_i and v_i, and $v_j \in V$. Note that neighbouring vehicle v_j defines the vehicle within the transmission range of vehicle v_i. 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). The vehicle supposes to has direct communication with its surrounding neighbouring vehicles at least once, evaluating the experience and assigning an assessed trust value to each neighbour (T_e). At some point, vehicles V start spreading reports about an event's occurrence on the road. Vehicle v_i receives reports from its preceding neighbouring vehicles v_j about an event's occurrence (E), 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 content (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 its 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 authorized and report form v_j is valid. We conclude the notations and descriptions in Table 3.. To establish trust between vehicles, we assume that vehicle v_i sends well-known answer question to the neighbouring vehicle v_j and based on the received answer (e.g., "Is there a traffic light ahead?"), trust is established and evaluated. Also, vehicle v_i can establish trust by sending a question to v_j about road safety and traffic (e.g., "Is the road clear?"). Then, based on the report's trustworthiness, the sender's trust (i.e., vehicle v_j) is evaluated (e.g., trust =1 if v_j is honest, and trust = 0 if v_j is dishonest). Also, we assume that the vehicle v_i receives conflicting reports (i.e., R, and R') regarding the same event E. We assume that in a case of an event such as a car accident, the equipped sensors in the vehicle will send a warning message to the surrounding vehicles within its transmission range.

3.2.2 Report Validation Module (RVM)

Due to the high mobility of vehicles in VANET and the size of communication, it is important to allow passing only 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 neighbour 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 [8] and [51] to identify the validity of the event (i.e., "Active" or "Expired"). Moreover, in [8], 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 [8]. In other words, the report relevancy in [8] is checked by ensuring that the sender is 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 to him. This may and 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 [75].

Table 3.1: Network Model Notations.

Notations	Description
v_i	Any vehicle in the network, i=1,2, 3,..
v_j	Surrounding neighbour 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
$t_{diff.}$	The time difference between $t_{curr.}$ and E_{st},
EL_{valid}	Valid events list
$EL_{invalid}$	Invalid events list

When event E is reported for the first time, v_i checks if the event is still active and relevant or not. 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 3.2. We construct two lists, called EL_{Valid}, $EL_{Invalid}$ to keep tracking any future reports regarding the same event. In other words, after E_d experis, the event is automatically moving from EL_{Valid} to invalid events list $EL_{Invalid}$.

Table 3.2: 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**

It is important to note the following: 1) the event duration E_d depends on the event type [8,51]. For a major event such as route 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 [8], 2) according to our assumption that 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 our scop in this work, 4) any report follows *Case 1* is added to EL_{valid} regardless of its content (i.e., agree with the event or not), 5) 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).

3.2.3 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 distrusted senders. For authentication purposes, we suggest using the authentication scheme proposed in [7] 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 [76] 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., $T_{thr.} > 0.6$ [77]). As shown in Figure 3.1 the report from the sender who 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. Note that the vehicle can determine n reports based on the task at hand.

3.2.4 Risk Estimation Module (REM)

By reaching this point, the receiver v_i received n^{th} contradictory reports (R and R') about an event's occurrence E from its neighbouring vehicles (i.e., report R informs that event E does exist, and report R' informs that event E does not exist). REM provides a decision-making process for vehicles facing conflicting reports. This is done by aggregating n^{th} 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 neighbours behind it. Vehicle v_i computes the risks for the associated action of believing R or R', then, the vehicle makes a decision to take the action with the lowest estimated risk. This module comprises two phases:1) Risk assessment phase and 2) Decision-making phase.

- ***Phase 1: Risk Assessment***

In this phase, risk estimation is integrated with a multi-dimensional trust model in [43] that is used experience-based, role-based, hop-based and majority-based. We assume that each vehicle has a direct communication (T_e) with its neighbours (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 [43]. Risk is defined in accordance with the USA National Institute of Standards and Technology (NIST) in [78], as follows:

$$Risk = Likelihood \times Impact \qquad \text{(Eq. 1)}$$

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 n^{th} 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 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 $\theta = \theta_{R'}$. 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* (*ERR1*) and a *Type II error* (*ERR2*). Using (Eq. 1), a separate risk function L can be defined for each action as:

$$L(a_R, \theta) = \alpha\left(\theta \,|a_R\right) M_\alpha \qquad \text{(Eq. 2)}$$

$$L(a_{R'}, \theta) = \beta\left(\theta \,|a_{R'}\right) M_\beta \qquad \text{(Eq. 3)}$$

where α ($\theta|a_R$), and β ($\theta|a_{R'}$), represents the likelihood of *ERR1*, given action a_R; and the likelihood of *ERR2*, given action $a_{R'}$, and M_α and M_β are the impacts associated with the *ERR1* and *ERR2* errors, respectively. In the following, we shorten the notation α ($\theta|\, a_R$) and β ($\theta|\, a_{R'}$) to α and β for convenience.

Before explaining likelihood score computing, we give a brief description of the main components of the likelihood computation formula. We use the same category in [47], where the vehicles are divided based on their relations with the event in three catagoryies:1)Event Reporter (the vehicle is involved in the event), 2) Event Observer (the vehicle witness the event and within one hop from the event reporter), and 3) Event Participant(the vehicle within two or more hops from the event report). For simplification purposes, we use the abbreviation for each category in [47] as follows: ER, EO, and EP. The likelihood score formula is combined between work in [43] and the weighting scheme in [19]. Move to hop-based model in [19], where the model aims to overcome cascading and oversampling issues 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 ($\alpha_{hop=1,2,..,n}$) multiply by the vehicle decision (i.e., d_j = 1if v agrees with the event; otherwise, d_j = -1). Then, the hops weights by the corresponding decision are aggerated (W_d). The event exists if W_d is

greater than 0; however, if W_d is lower than 0. 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 computing, 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 [43] by integrating the weight of each report based on the number of hops from the event, as follows:

$$E(R_j) = \sum_{v_j \in V} W(R_j) \left[\frac{T_e(v_j) T_r(v_j)}{C_t(v_j) C_l(v_j)}\right] \quad \text{(Eq. 4)}$$

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:

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

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:

$$\alpha = 1 - \beta = \frac{E(R')}{E(R)+E(R')} \quad \text{(Eq. 6)}$$

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 *ERR1* and *ERR2* errors. We defined a factor called the error intensity that presents the measure of the vehicle damage size due to *ERR1*, and *ERR2* errors, respectively. Also, we propose 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:

$$\frac{M_\alpha}{M_\beta} = \frac{I_\alpha}{I_\beta} \quad \text{(Eq. 7)}$$

where M_α/M_β is the risk ratio.

As earlier mentioned, *ERR1* 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 θ_R'. The driver's action a_R could be slowing down the speed, changing the lane, or entering the nearest exit. The type of action a_R and *ERR1* error intensity (I_α) is dependent on the number of vehicles that are affected by others' decisions and the proximity of the vehicle to the reported event. Let vehicle v_i on the highway, and the driver receives a notification about an accident within T time from its location, and there are N neighbours following the vehicle (i.e., may affect 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 *ERR1* 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 *ERR1* error is. On the other hand, the larger N is the larger error intensity is. Thus, we model the error intensity I_α of *ERR1* as:

$$I_\alpha = a + (N/T)^b \qquad \text{(Eq. 8)}$$

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 *ERR2* 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 break suddenly or swerve. Thus, the immediate consequence of the *ERR2* 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 neighbouring, the impact of *ERR2* increases too (i.e., more vehicles are affected). Thus, we also interpret I_β to have a form similar to I_α , as follows:

$$I_\beta = c + (N/T)^d \qquad \text{(Eq. 9)}$$

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 $L(a_R, \theta)$ and $L(a_{R'}, \theta)$ of both actions are calculated using (Eq. 2) and (Eq. 3).

Note that: 1) the number of affected neighbouring vehicles (N) should be at least 1, 2) the number of neighbours following the vehicle *N* is directly proportional to *I* and *T* is inversely proportional to *I*, and 3) the shape of *ERR1* and *ERR2* intensity curves could take different shapes according to the road conditions (i.e., different scenarios). In Section 3.2.5, we explain the three possible curves of *ERR1* and *ERR2* intensity. Also, the scenario mentioned above is one of these scenarios, called RTEAM-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, θ) and $L(a_{R'}, \theta)$. Thus, the Bayes' decision rule is used [79] 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) [43,80] and the new $T_{e(j)}$ is updated according to [8] as follows:

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

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 an expire event) or giving false event state, the sender's trust value is updated (i.e., trust value is decreased) [43,80] and the new $T_{e(j)}$ is updated according to [8], 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):

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

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

3.2.5 Impact 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 an unexpected event occurring (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 *ERR1*, and *ERR2* 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 *ERR1*, and *ERR2* errors as shown in Figures 3.2,3.3, and 3.4, respectively. The impact of both types of errors is increased by time. However, the impact of *ERR2* is always larger than the impact of *ERR1* when the vehicle is too close to the event (i.e., by time decreasing) and no action is taken yet.

- RTEAM-1 is shown in Figure 3.2 where *ERR2* error is always higher impact than *ERR1* 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 *ERR2* (i.e., impact) is always larger than the intensity of *ERR2,* and both impacts are increased the time.

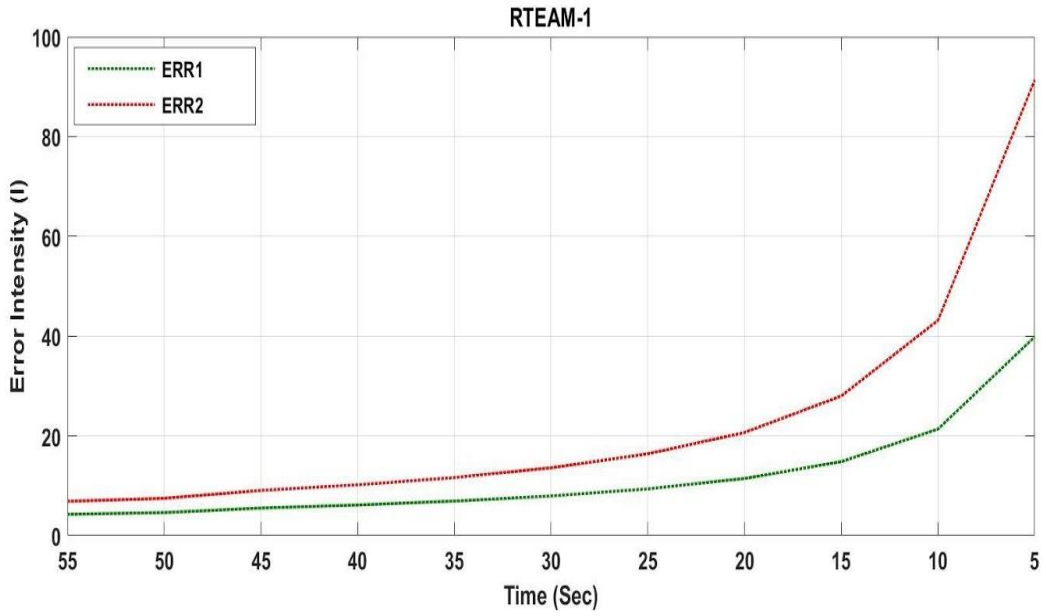


Figure 3.2: *ERR1* and *ERR2* curves of RTEAM-1.

- RTEAM-2 is shown in Figure 3.3, where the intensity of *ERR2* error (the dashed red curve) is found to be lower than the *ERR1* (the dashed green curve) at high *T* because, with *ERR2* error, the drivers just keep carrying on without disruption. The drivers have time to make the right decision. Somewhere around *T* = 20sec, the *ERR2* error starts to become a bigger problem than the *ERR1* 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 on highways, urban areas with high density, or urban areas in rush hours. In other words, this scenario reflects the case where the intensity of *ERR1* (i.e., impact) is larger than the intensity of *ERR2* at high *T,* and when *T* is decreased (i.e., the event is closed), *ERR2* is gradually increased until reaching some point where the curve jumps (i.e., the event is too close).

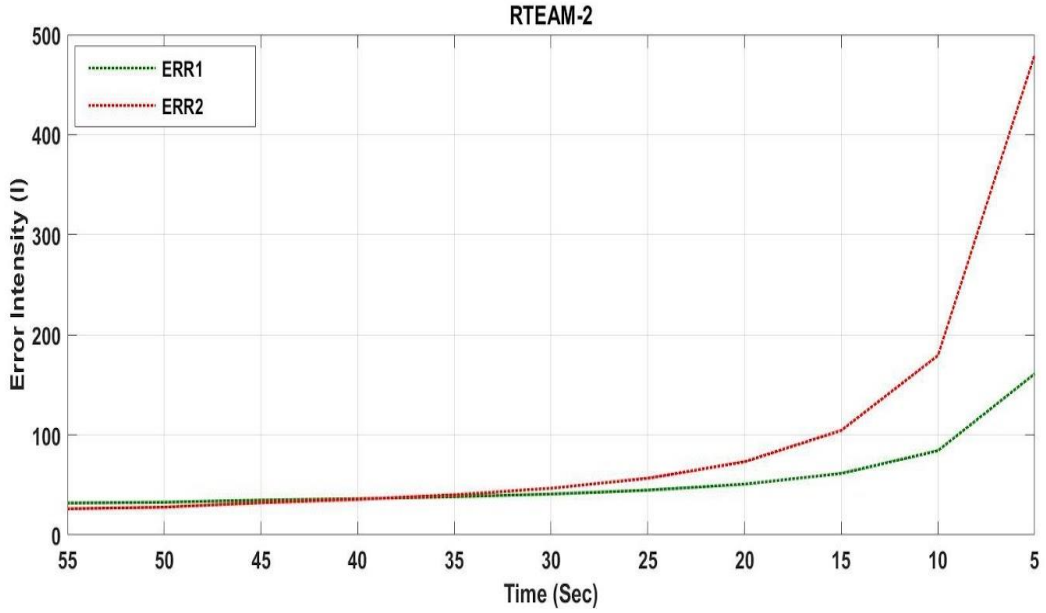


Figure 3.3: *ERR1* and *ERR2* curves of RTEAM-2.

- RTEAM-3 is shown in Figure 3.4, where both errors have the same impact at high *T*. Then, by the time passing, *ERR2* 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, *ERR2* impact gradually increases.

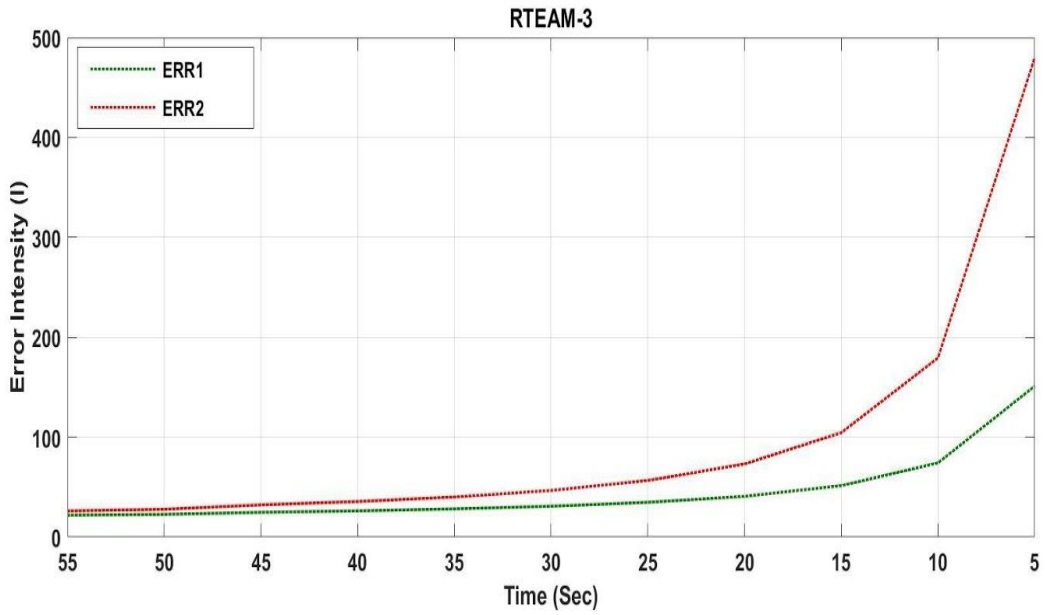


Figure 3.4: *ERR1* and *ERR2* curves of RTEAM-3.

3.3 Illustrated Example

In this section, we demonstrate an example scenario to go through the calculations of Section 3.2.4. This example aims to show that the purely trusted-based methods may take the wrong action in some cases compared to our model. To show the effectiveness of our work, we compare our model (RTEAM) with:1) Simple Trust Model (STM) [75] and 2) Multi-faceted Trust Model (MTM) in [43]. The STM is simple voting on an event method that considers the majority of voters on the event are trusted. For example, if the majority of the vehicles in the network agree with the event's occurrence, then the output of STM is "Event exists".

On another side, multi-faceted trust model (MTM), where each report sender is weighted based on its experience-based, role-based trust and its time and location closeness of the voter. Finally, the event state is decided (i.e., "Exists" or "Does not Exist") according to n^{th} reports. If the majority consensus has been reached, the majority's opinion is followed; otherwise, the opinion of the most trusted vehicle (i.e., the highest experience-based and role-based trust) is followed.

We assume n vehicles on a highway where each vehicle has a previous direct with other vehicles and all vehicles have the same role (i.e., all vehicles are ordinary vehicles). We consider a scenario where vehicle v_{11} receives votes (i.e., reports) from other vehicles on an event's occurrence (i.e., an accident), as shown in Figure 3.5.

Assume that vehicle v_{11} has N neighbouring vehicles (i.e., the vehicles behind v_{11}) and $N = 6$. Suppose at T= 50 sec, v_{11} receives conflicting reports from the vehicles v_1 through v_{10} about an event (i.e., car accident), as identified in Table 3.3 along with their role-based and experience-based trust, location closeness, time closeness, and the number of hops attributes.

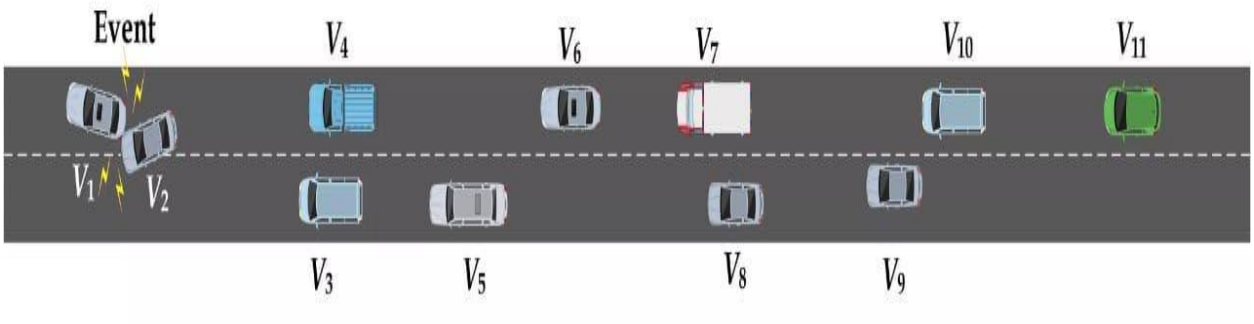


Figure 3.5: Car accident Scenario.

Table 3.3: The example Information about vehicles.

Vehicle ID (v_i)	T_e	T_r	C_l	C_t	E(R)	Report Category of the sender	Num. of hops
v_1	0.80	0.5	0.1	0.1	40	**R** Reporter	**0**
v_2	0.80	0.5	0.1	0.1	40	**R** Reporter	**0**
v_3	0.80	0.5	0.2	0.2	10	**R** Observer	**1**
v_4	0.85	0.5	0.2	0.2	10	**R'** Observer	**1**
v_5	0.80	0.5	0.2	0.2	10	**R** Observer	**1**
v_6	0.80	0.5	0.4	0.4	2.5	**R'** Participate	**2**
v_7	0.80	0.5	0.5	0.5	1.6	**R'** Participate	**3**
v_8	0.80	0.5	0.5	0.5	1.6	**R'** Participate	**3**
v_9	0.80	0.5	0.5	0.5	1.6	**R'** Participate	**4**
v_{10}	0.80	0.5	0.7	0.7	0.81	**R'** Participate	**4**

First, assume that substantial data has been collected about the outcomes of *ERR1* and *ERR2* errors with respect to highway accidents, resulting in error intensity curves defined according to (Eq. 8) and (Eq. 9), as follows:

$$\mathrm{I}_{\alpha} = 1 + \left(\frac{\mathrm{N}}{\mathrm{T}}\right)^{0.9}, \quad \text{and} \quad I_{\beta} = 1 + \left(\frac{N}{T}\right)^{1.1}$$

We assume that all ten vehicles have the same experience-based and role-based trust except v_4 to serve case two of MTM, where the model relies on the highest trusted vehicle's opinion. According to the nature of attacks in VANET, at some point, any legitimate vehicle could misbehave and turn to be malicious. Even though v_4 is the highest trusted vehicle but it might change its behaviour and lies about an event's occurrence. Using STM, the majority opinion is report *R'* (i.e., six vehicles out of ten send report *R'*). Thus, v_{11} will take the associated action $a_{R'}$, which is the wrong action. Using MTM, the aggregated effect from the reports (i.e., The report confirms the existence of the event) sent by vehicles v_1, v_2, v_3, andv_5 is calculated as follows:

$$\mathrm{E}(R) = 2 \times \left(\frac{0.8 \times 0.5}{0.1 \times 0.1}\right) + 2 \times \left(\frac{0.8 \times 0.5}{0.2 \times 0.2}\right) = 100$$

The aggregated effect from report *R'* sent by vehicles v_4, v_6, v_7, v_8, v_9 andv_{10} is calculated as follows:

$$\mathrm{E}(R') = \left(\frac{0.85 \times 0.5}{0.2 \times 0.2}\right) + \left(\frac{0.8 \times 0.5}{0.4 \times 0.4}\right) + 3 \times \left(\frac{0.8 \times 0.5}{0.5 \times 0.5}\right) + \left(\frac{0.8 \times 0.5}{0.7 \times 0.7}\right) = 18.73$$

According to [43], the maximum accepted error for v_{11} is 0.10. The majority consensus is calculated as follows:

$$\text{Majority-Consensus} = E(R)/ [E(R) + E(R')] = 0.84 < 0.9$$

Based on MTM, the majority consensus is not reached. Therefore, v_{11} will believe the report from the highest trusted vehicle. Thus, the vehicle with the highest trust is a vehicle that votes with report *R'* (i.e.v_4), then, v_{11} believes report *R'* and takes the wrong action $a_{R'}$.

Using the risk-based approach (RTEAM), the aggregated effect of the reports that are confirming the existence of the event) sent by vehicles v_1,v_2,v_3, andv_5 is calculated according to (Eq. 8), and we get E(R) = 440. The aggregated effect of report *R'* sent by vehicles v_4, v_6, v_7, v_8, v_9 ,and v_{10} is calculated, and we get E(R') = 40.49.

Therefore, the likelihood scores are calculated as follows:

$$\alpha = 1 - \beta = \frac{E(R')}{E(R)+E(R')} = 0.084,\ \beta = 0.92,\ \text{and}\ \beta/\alpha = 10.95$$

Calculating the risk ratio according to (Eq. 8) and (Eq. 9),

$$\frac{M_\alpha}{M_\beta} = \frac{1 + (\frac{N}{T})^{0.9}}{1 + (\frac{N}{T})^{1.1}} = \frac{1 + (\frac{6}{0.83})^{0.9}}{1 + (\frac{6}{0.83})^{1.1}} = \frac{6.93}{9.8} = 0.7$$

The risk ratio is smaller than β/α, then, v_{11} will believe the report *R* with the lowest risk (i.e., $L(a_R, \theta) = 0.58 < L(a_{R'}, \theta) = 9.01$). Therefore, action a_R is maintained, and the neighbouring drivers are notified about the event.

3.4 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 [81]) 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 excess 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 *ERR1*, and *ERR2*

errors with respect to highway accidents, resulting in the error intensity curves of (Eq. 8) and (Eq. 9) to be defined according to the values of *a*, *b*, *c*, and *d* as shown in Table 3.4.

Table 3.4: 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
Vehicles 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

To show the effectiveness of RTEAM, we compare RTEAM with 1) Simple Trust Model (STM) , where the earliest report is followed, 2) Hop-based Trust Model (HTM) [19], and Multi-faceted Trust Model (MTM) [43].

We defined the following metrics to evaluate the efficiency of RTEAM-1, RTEAM-2, and RTEAM-3 in terms of security.

- 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.

- RISK LEVEL (RL): the risk level (RL) corresponding to *ERR2* (when a vehicle detects no event even though there is an event).

A. 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 3.6 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 that 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 the decision in the way that the vehicle cannot identify the event state (i.e., the weight of the vehicles agree with the occurrence's of the event mines the weight of the vehicles disagree with the occurrence's of an event is equal zero). With increasing the malicious in the network (i.e., most of the participants deny the occurrence's of the event), the UND cases are decreased (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 the action regarding an event state. Regarding 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.

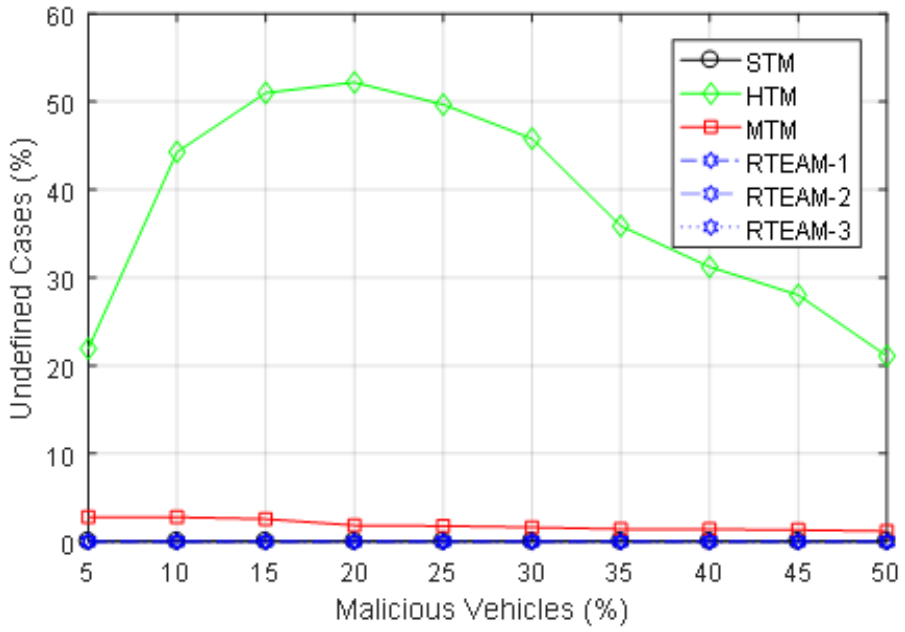


Figure 3.6: Undefined Cases (UND).

B. 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 attackers gives a high chance for the false event state to spread around the network, which negatively affects the other vehicles' decisions.

As depicted in Figure 3.7, the TPR of STM, HTM, and MTM emphasizes that the network attains lower TPR compared to RTEAM-1, 2 and 3. HTM and MTM show slightly similar TPR with different malicious injections. With a network injected with 5% malicious vehicles, RTEAM achieves higher TRP (about 88%) compared to HTM and MTM, where TRP is less than 80%. This is due to the fact that RTEAM inherits the advantage of HTM, and MTM (i.e., Multi-faceted trust and hop-based trust are incorporated in RTEAM). The gap between RTEAM-3 and HTM and MTM is about 20%, and it is slightly increased to be 30% to 40% higher when the number of malicious vehicles is between 10% and 25%. Moreover, RTEAM-3 can achieve higher TPR than STM. RTEAM-3 outperforms all trust models and even the scenarios of 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 3.7 clearly depicts that RTEAM can achieve higher TPR and get better results than other trust models, especially, the number of injected attackers is 25%. Even though RTEAM show low TPR when the percentage of malicious is increased (due to the fact that HTM and MTM are parts of its architecture), but 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 nodes around the network are malicious nodes.

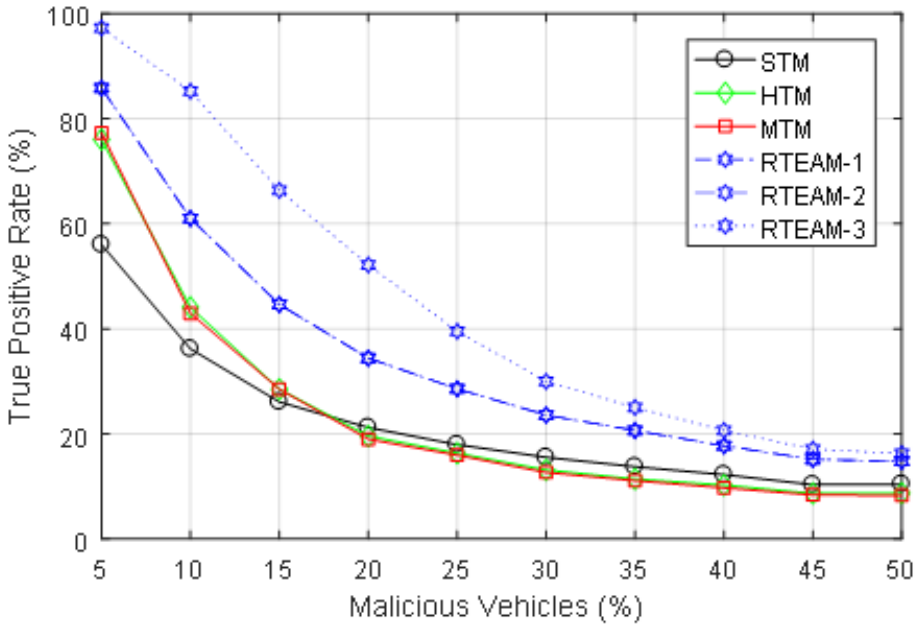


Figure 3.7: True Positive Rate (TPR).

C. RISK LEVEL (RL)

The Risk Level (RL) is computed based on the average of the associated risk corresponding to the occurrence of *EER2*. As shown in Figure 3.8, the risk level (RL) is increased by increasing the number of attackers in the network. STM, HTM, and MTM are purely trusted-based models that neglect the associated risk of the driver's action, where is this action may affect the safety of other drivers. MTM and HTM show better RL compared to STM with a smaller number of malicious nodes in the network (e.g., less than %20). However, with increasing the number of malicious nodes (more than %20), STM shows bad RL compared to MTM and HTM. RTEAM, with its three scenarios, outperforms the multi-faceted trust model (MTM) since it aims to take action with the lowest possible risk.

RTEAM provides better results than the other models, with the percentage of malicious between 5% up to 50% due to the fact that it mainly targets action with the lowest risk level. Among RTEAM scenarios, we notice that RTEAM-3 outperforms the other two scenarios (RTEAM-1 and RTEAM-2), which indicates that RTEAM could be more effective if it is used in urban areas compared to other areas such as highways.

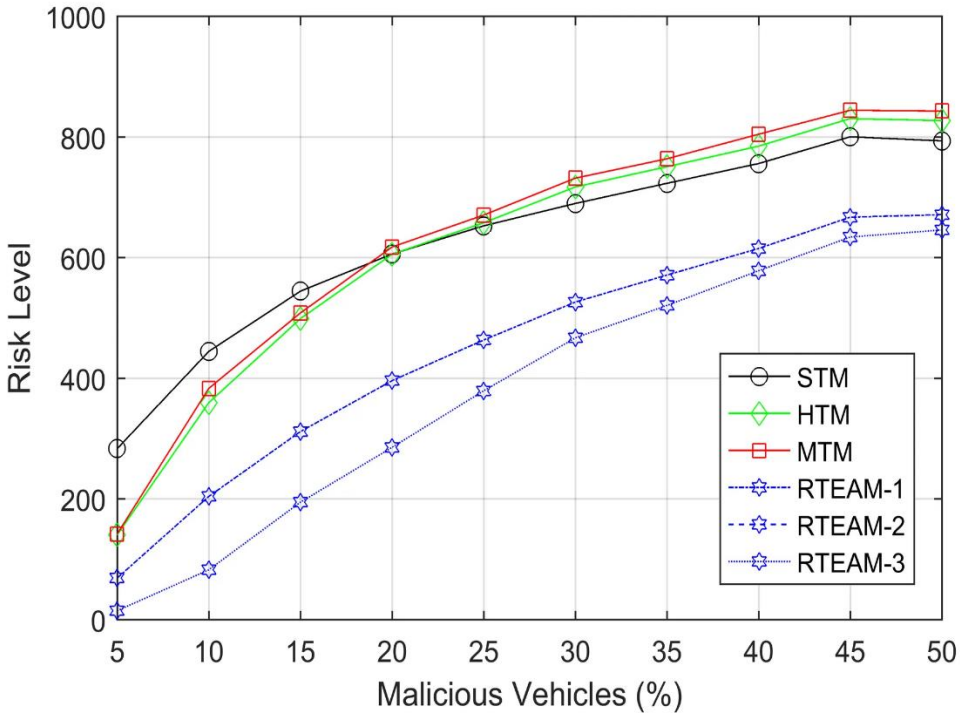


Figure 3.8: Risk Level (RL).

3.5 Discussion

In this section, we discuss RTEAM in terms of limitations and possible modifications. 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 makes 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 designing based on trust metrics to compute its likelihood, which affects its performance. Another limitation is that RTEAM cannot process the case where we only have a single in front neighbour (i.e., only one 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. In other words, RSU can check its active event list before confirming the event to the vehicle.

Other possible modifications can improve the performance of RTEAM as follows: 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) is such as the model in [51], and 2) adding a Payment Punishment Scheme (PPS) such as in [42] to encourage vehicles to participate in voting on the events in the network.

3.6 Conclusion

In this chapter, we have proposed a risk-based trust model (RTEAM) for VANETs. The proposed model improves the decision-making process by integrating risk estimation into the trust evaluation process of incoming reports. Simulations 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 chapter, 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 implementing more comprehensive simulations using different scenarios in addition to comparing against other existing models.

Chapter 4:

FREM: Fog-based Reputation Evaluation Model

4.1 Introduction

Two special, predominant features of the VANET are as follows: 1) the VANET is an ephemeral network in which the topology rapidly changes due to the high speed of the vehicles (e.g., up to 120 km/hr.), and 2) the VANET is a large-scale network where the number of connected nodes is high. So, at high speed, the vehicles do not have a long time to contact each other to establish and evaluate trust. Therefore, efficient and reliable trust evaluation in VANETs is an arduous challenge.

Another important issue encumbering most of the existing trust evaluation models is their reliance on real-time experience evaluation (i.e., experienced-based trust), which is erased from the system rather than forming an insightful record for future evaluations. The logic behind neglecting reputation (i.e., past evaluation records) and considering only real-time experience trust is to conserve capacity (i.e., vehicle's memory) and resources (e.g., cost of infrastructure such as RSUs) required for gathering and retaining extensive reputation records around the city. Relying on a real-time experience busies the node with communication (i.e., for evaluation purposes), yet the node may not even use this evaluation for any imminent interaction. For instance, vehicle v_i contacts vehicle v_j to evaluate its trust, and during the communication session, v_i does not use this evaluation for any purpose (i.e., communication resources are spent for no gain).

"All in the same boat" is an issue in the existing reputation systems that we explore in this chapter. We define the "All in the same boat" issue as the situation where a reputation evaluation algorithm aggregates the reputation of different experience types for a given node in a single reputation value. For example, reputation updating in [82] is computed as an aggregated value that includes all evaluations of the past experiences regardless of the experience types, repetition, and effect on the network. To illustrate, suppose a node performs well in replying to three non-safety applications requests (e.g., its reputation for these requests is 3 out of 3) but performs with mediocrity in replying to three safety-applications requests (e.g., its reputation for these requests is 2 out of 3). In this case, the node will gain a high reputation value (i.e., 0.83) even though it is likely to have caused a harmful effect on the network (e.g., major accident) when it responded inappropriately to one of the safety applications requests. Thus, the traditional reputation systems may blindly employ selfish and malicious nodes to perform sensitive tasks that may affect road safety based on single, aggregate reputation values. We call this "Blind Employment" and define it as the

case where a vehicle or infrastructure relies on a node with the highest reputation value without considering the type of past experiences.

Different reputation updating algorithms, such as [7, 8, 51, 82], may all suffer from the issue of "All in the same boat". In addition, the evaluations in these models may not necessarily be related to the time of experience (i.e., time of evaluation). For example, in [7, 51], the time decay of the experience is neglected, and only reward and punishment mechanisms are applied to update the reputation. Conversely, more recent experience is weighted more heavily than old experience in [8] (e.g., Eq. 11 in Chapter 3), in addition to rewarding and punishment mechanisms. We provide more details about [51] and [82] when we explain Task-based Experience Reputation (TER) in Section 4.3 to show the "All in the same boat" issue.

As mentioned in Section 2.4, fog computing has been introduced to support the VANET environment and enhance the quality of the provided services. Fog computing extends conventional cloud computing to the edge of the network [17] and holds the promise of significant potential benefits to edge users. Situated as an intermediate layer and distributed geographically throughout the VANET landscape, fog can benefit vehicle-based applications in terms of response time, communication, and storage [29]. The capabilities of fog and its position (i.e., the proximity from edge users) endow fog with the power to play a vital role in employing the most competent node (i.e., the node with the highest TER). Moreover, fog can be used to accurately detect events on the road and propagate the event's details to the vehicles of the VANET, as we will see in Chapter 5. In other words, fog can reduce the workload that is required of the vehicles (e.g., propagating the event's details, evaluating the trust of the sender). In [83], the authors suggest using fog nodes to support trust management in VANETs. Therefore, in this work, we deploy fog nodes to gather the trust evaluations from the vehicles, which allows fog nodes to rely on their local vehicles to perform certain tasks. Also, fog nodes are used in this work to maintain the records of its local vehicles, thereby reducing the need to communicate with the cloud.

In this chapter, we aim to construct a fog-based reputation evaluation framework in a way that meets the following: 1) leveraging past experience evaluations as rich sources of information that may be referred to in subsequent trust evaluations (i.e., the evaluations reflect the node's past), 2) designing a fog-based reputation evaluation system to support ITS such that infrastructures bear the majority of the workload, 3) respecting economic feasibility by avoiding expensive infrastructure such as RSUs (i.e., fog is a good alternative), 4) preventing the "All in the same boat" issue by utilizing the concept of (TER), and finally, 5) demonstrating the value of TER in realizing the "smart employment" concept in VANETs.

4.2 Proposed Model Architecture

In this section, we describe the main architecture of our model and the function of each part. We illustrate the use of a vehicle's historical trust records (reputation) as an alternative to real-time trust records. As shown in Figure 4.1, our proposed model contains three basic layers: 1) Edge layer (i.e., vehicles and IoT devices), 2) Intermediate layer (i.e., Fog nodes and RSU), and 3) Main layer (i.e., Cloud and Trust Server).

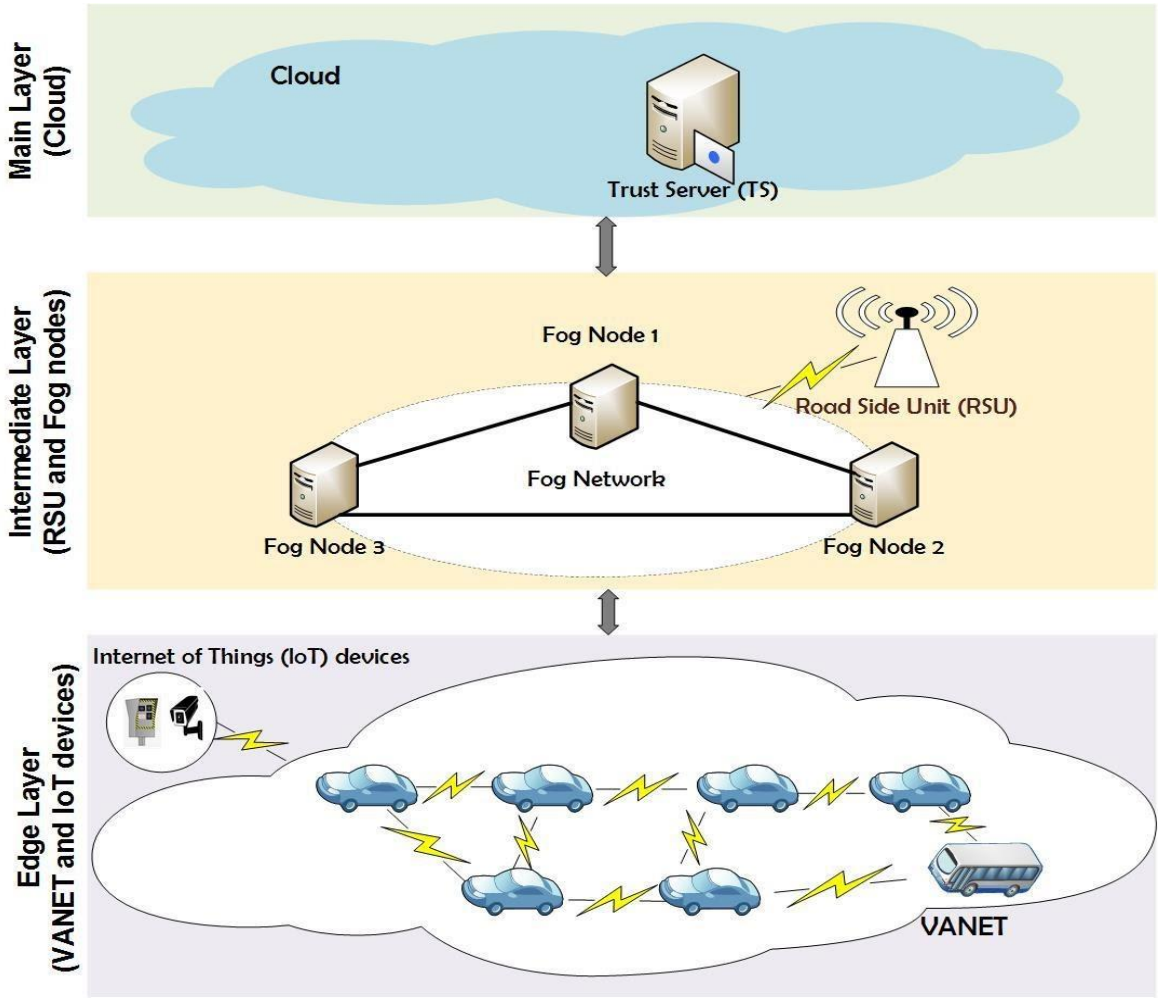


Figure 4.1: Fog-based Reputation Evaluation Model (FREM) architecture.

- ***Vehicles***: Vehicles are the mobile component of the VANET environment. We assume that any vehicle registered on the ministry of transportation system has the following main attributes:

  1) Vehicle ***Identification Number*** (**ID**): a unique identification number issued by a trusted party (e.g., Ministry of Transportation).

2) Vehicle ***Digital Trustworthiness Card*** (**DTC**): a digital card is identified by a unique identification number and issued by a trusted party (i.e., CA) and contains the vehicle ID, the predefined role (T_r), Task-based Experience Reputation (TER) of a vehicle for specific tasks (i.e., multiple TER values).

The (TER) values are visible to the driver (i.e., for transparency purposes) and are sharable. The vehicle role (T_r) may be one of the following: 1) governmental vehicles and public transport vehicles, and 3) ordinary vehicles. TER_i values reflect the node reputation of doing a certain task S_i and are stored on the DTC in tuple format (i.e., (TER_1, TER_2, .., TER_n)) on a server, called the Trust Server (TS). The TER values are collected directly by fog nodes after the vehicle performs any direct interaction with its peers; however, updates to the DTC of TER values may take up to 24h.

- ***RSU, Fog nodes,*** **and** ***TS*****:** RSU and fog nodes form geographically distributed infrastructures that provide elastic resources and services to vehicles of the edge layer. Each RSU oversees a geographical area that includes a set of fog nodes sequentially connected under its umbrella. Fog nodes are spatially distributed such that each is located within the transmission range of its neighbouring fog node. The RSU has direct access to the TS in the cloud and can update and retrieve DTCs of vehicles on the road.

For better understanding, let us consider the following scenario. Vehicles v_1, v_2, v_3, and v_4 have used the road located in Area_1, which is covered by RSU_1 and its fog nodes (i.e., Fog_1 and Fog_2). Each vehicle communicates with neighbouring vehicles directly, evaluates the experience, and assigns to each an assessed trust value (T_e). Assume that v_1 has direct experience with v_2 (e.g., v_1 follows v_2's advice) in Area_1. Vehicle v_1 evaluates its experience with v_2 and sends its evaluation to the nearest fog node before leaving the area. The evaluation message contains the peers' IDs, the experience stamp time, trust value, and the task type as (v_1^{ID}, v_2^{ID}, t, T_e, S_i), which is called Trust Evaluation Message (TEM). Each trust value must also identify the vehicle number and experience stamp time to prevent duplication of information. Fog nodes collaborate to aggregate the collected trust values (TERs) and send them to the RSU. The RSU sends the latest update to the TS. Later, the TS aggregates a vehicle's DTC records that are recorded in various areas (i.e., different areas from different RSUs around the city). It is worthy of note that fog nodes store these records in their memories until the next update from the RSU is received. The RSU also keeps these records until it receives the updated records from the TS.

➢ *Proposed model goals*

Our design takes into consideration the following points:

1) ***Decentralization***: The TEMs are collected by fog nodes and sent to a preselected fog node. The preselected fog node (i.e., master fog) is responsible for extracting TER values from TEM for each vehicle and then filtering these TER values. Then, the master fog sends a list of vehicles with their TER values to the RSU. Next, the RSU at time t forwards

this list to the TS server. Finally, TS updates RSUs with the updated list (i.e., the vehicles with the new TER). So, the TER is calculated based on the cooperation of all the nodes in the network (i.e., vehicles, fog nodes, and RSUs).

2) ***Security***: We assume the TS, RSUs, and fog nodes are trusted parties and the communication between them is secured. Moreover, we assume that DTC records are updated only by the trusted, authorized party (i.e., the RSU and the TS) in the intermediate layer each time a new trust evaluation is obtained for the vehicle. RSUs are responsible for ensuring that the DTC of a vehicle is a real/valid certificate by checking that the certificate ID exists on its database (for local vehicles) or the TS database (for newcomers vehicles). Moreover, the RSU checks that the DTC records are not manipulated, which can be verified in two ways: 1) for the well-known vehicles (local vehicles), the RSU compares the DTC with the latest DTC records (i.e., TER values) retrieved for the same vehicle -- any suspicious change (i.e., a high increase in TER value) prompts the RSU to send a request to the TS to retrieve the DTC records for the vehicle, 2) for newcomers vehicles: the RSU sends a request to the TS to retrieve the DTC records. In both situations, the RSU can temporarily suspend any vehicle from communicating with others until it verifies the DTC and TER values. In other words, any vehicle with a fake DTC, invalid DTC, manipulated DTC records or TER value below the network trust threshold is suspended from performing tasks having thresholds defined by these parameters (e.g., TER for task S_1 = 0.6 and trust threshold is 0.7; the vehicle is suspended from performing task S_1). Besides, the RSU inserts a special flag (i.e., a flag that is changed periodically, perhaps a colour, a special character, a number, or a word) in its welcome message. Using the flag in every communication informs fog nodes and vehicles whether this node is fully or partially trusted. Finally, master fog utilizes an algorithm that eliminates any repeated TEM and/or fake/selfish evaluations (e.g., bad-mouthing).

Note: We assume the RSU changes the flag from time to time and the optimal time to do so is when the sensors on the road do not detect any newcomer vehicles (i.e., the road is empty). In addition, the RSU must inform its fog nodes about this change. Partially trusted vehicles can therefore be detected from their flags. For example, if the receiver flag is “Green” and the sender’s flag (i.e., the malicious vehicle) is “Red”, this indicates that the sender is not fully trusted.

3) ***Fairness***: since the VANET is a very busy environment and the mobility of its nodes is high, we focus on fairly dividing the workload between the infrastructure (i.e., fog nodes).

4) ***Timeline***s: The TER values may change with time (i.e., increase or decrease); thus, the updated values should be available to the RSU in good time. Also, the proposed model reduces the workload of the vehicles through conservative management of vehicle resources and time (e.g., time is not lost in the trust establishment phase).

➢ ***Proposed Model main operations***:

Three primary operations take place according to the proposed architecture, described as follows:

1. Local Storage of DTC:

A vehicle may travel the same path repeatedly during the day or the week as part of a routine commute. We call this path a ***Preference Path*** (PP). On the other hand, the same vehicle may also travel a new path, which we call ***Exploring Path*** (EP). The RSU stores each vehicle's DTCs according to the type of path that the vehicle is travelling through (i.e., vehicles are divided into two lists). Thus, the RSU classifies and stores the vehicles according to their travelling patterns with path *P* into two classes as follows:

- **Class A (Frequent visitors / Preference Path)**:

The RSU and fog nodes in the same area may be repeatedly visited by a vehicle, in which case it is defined as a frequent visitor of path *P*. Vehicles are classified as Class A when the time-frequency of visits to path *P* reaches a threshold (t_f). The Class A designation of the vehicle (e.g., vehicle v_i) is forgotten if v_i does not again travel path *P* within a certain time threshold. The RSU stores the Class A vehicles list in its long-term memory and updates the list periodically.

- **Class B (Occasional visitors / Exploring Path):**

Class B vehicles are vehicles that have visited an area in the last 13 days and are not on the Class A list. Class B vehicles are newcomers vehicles exploring the area (i.e., vehicles are travelling an EP) and may or may not revisit the same area in the near future. The RSU stores Class B vehicles in its short-term memory (i.e., temporary storage) and updates the Class B vehicles list every two weeks. Any Class B vehicle that meets the Class A definition is moved to the Class A list (i.e., to long-term memory). On the other hand, a vehicle is removed from short-term memory if it has visited less than *n* times in the last two weeks (e.g., less than five times).

The classification mainly aims to save the time of retrieving the DTCs for Class A vehicles from the TS. Also, by using this classification, we permit the infrastructure to rely on well-known vehicles instead of newcomer vehicles in performing sensitive tasks.

2. DTCs Records Updating:

Fog nodes, the RSU, and the TS update their databases (i.e., add/remove vehicles, or update DTC records) as follows: 1) **Vehicle-2-Fog** - at time t_0, fog nodes receive TEM message from a vehicle (e.g., v_1) about its neighbouring vehicle (e.g., v_2) on the road; 2) **Fog-2-Fog** - at time t_k, fog nodes send the list of recently active vehicles with their related information (i.e., the evaluator id, time of experience, trust evaluation, and task type) to the master fog node; 3) At time t_{k+1}, master fog

aggregates TEMs and extracts TER records from them. Then, the master fog filters (i.e., eliminates) malicious and duplicate records for each vehicle; 4) **Fog-2-RSU** - at time t_{k+2}, master fog sends the latest updated list (two separate virtual lists), which contains the vehicle ID and new TER values to the RSU; 5) **RSU-2-TS** - at time t_n, the RSU updates DTCs records for vehicles on Class A and Class B lists. Then, the RSU sends the list (i.e., the combined list of Class A and B) to the TS to retrieve the DTC records corresponding to the revised vehicles list. Note that t_n is in units of hours (e.g., 6 hrs.); however, if any node misbehaves, the RSU can send its list before reaching t_n; 6) **TS-2-RSU** - at time t_{n+1}, the TS updates the RSU with the current DTC records; and finally, 7) **RSU-2-Fog** - at time t_{n+2}, the RSU updates its database and shares the new updates with its fog nodes (see Figure 4.2). Note that any update to TER by the TS will directly appear in the vehicle's DTC records.

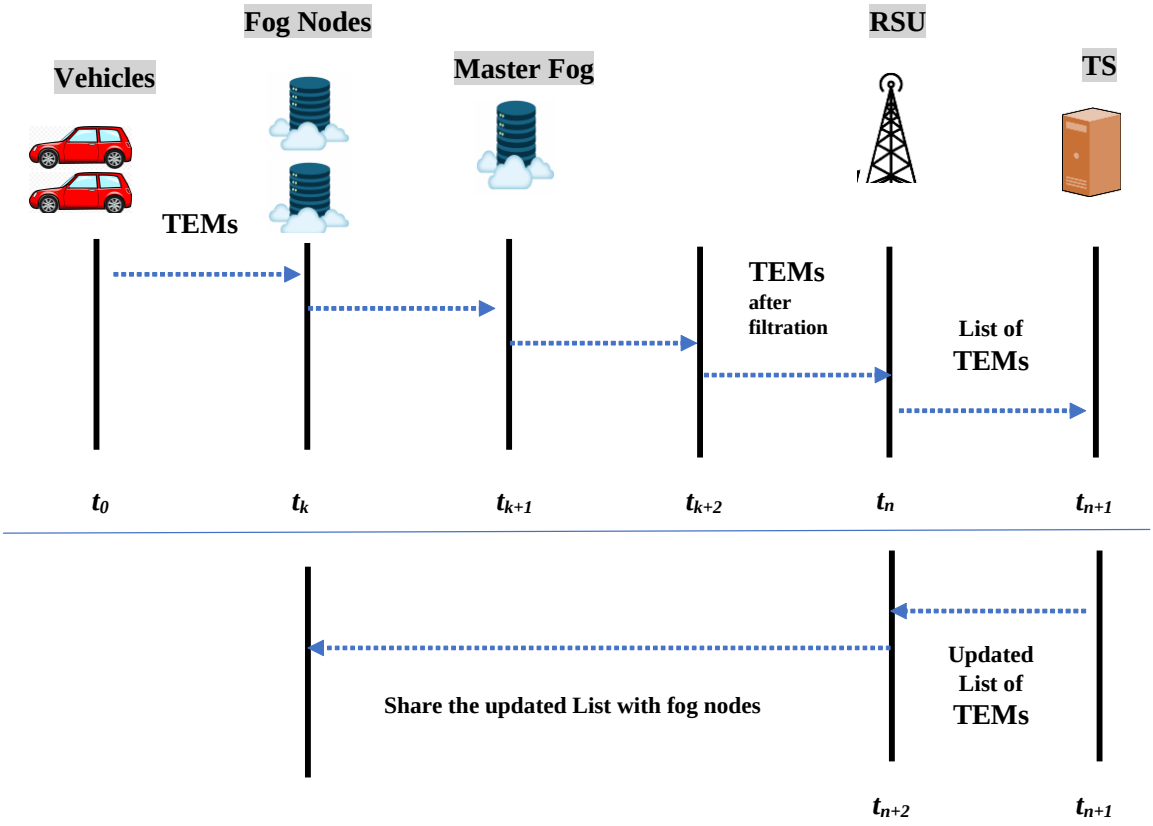


Figure 4.2: Digital Trustworthiness Card (DTC) records updating.

3. DTCs Retrieving:

All vehicles visiting an area send a “Hello” message containing its DTC to the RSU. If a visiting vehicle is not on the Class A or B lists, the RSU retrieves the vehicle’s DTC from the TS. To reduce the workload and latency of DTC retrieval by the RSU, we propose two retrieval methods as follows:

i) ***Offloaded Retrieval*** (for Class A and B vehicles)

In our model, the RSU stores DTC records of Class A and B vehicles in its long-term and short-term memory, respectively. When a vehicle sends a “Hello” message to the RSU, the RSU retrieves the most recent DTC records of this vehicle from its local memory, obviating the need to communicate with the TS. The RSU replies to the “Hello” message by sending a “Welcome” message with a flag. This flag should be used when the vehicle communicates with fog nodes or with other vehicles. Offloaded retrieval thus reduces communication between the RSU and the TS and saves time. We therefore promote offloaded retrieval, shown in Figure 4.3, as an effective and streamlined solution for the trust management of Class A and B vehicles.

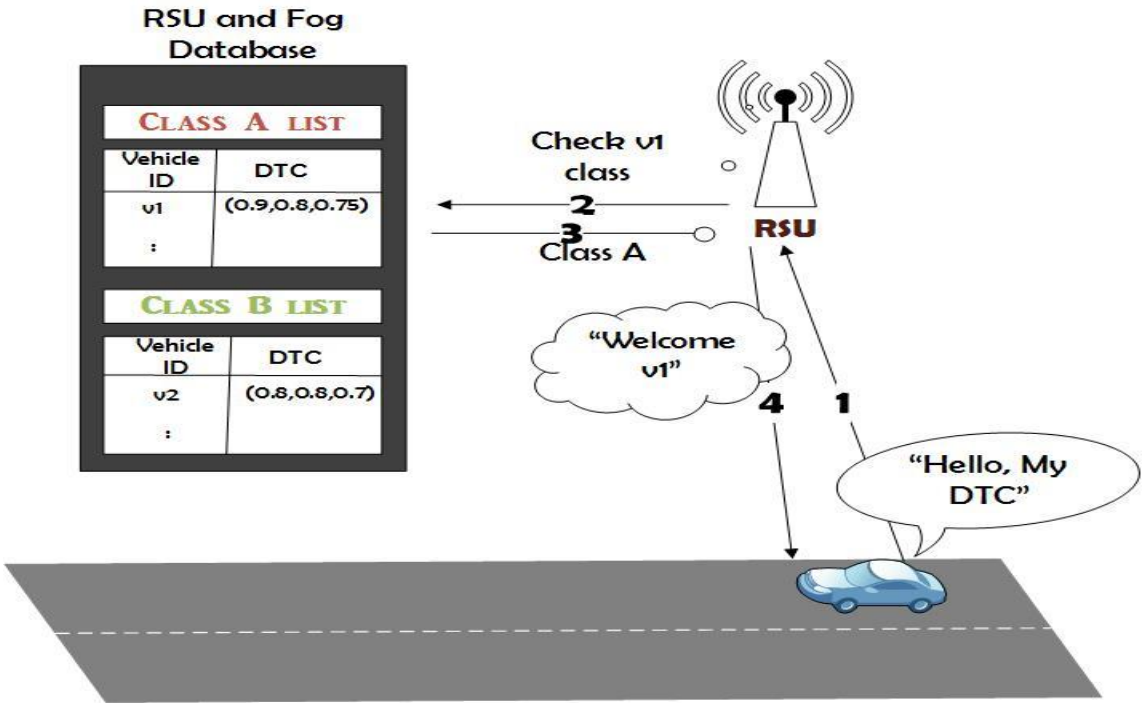


Figure 4.3: Offloaded Retrieval for Class A/B vehicles.

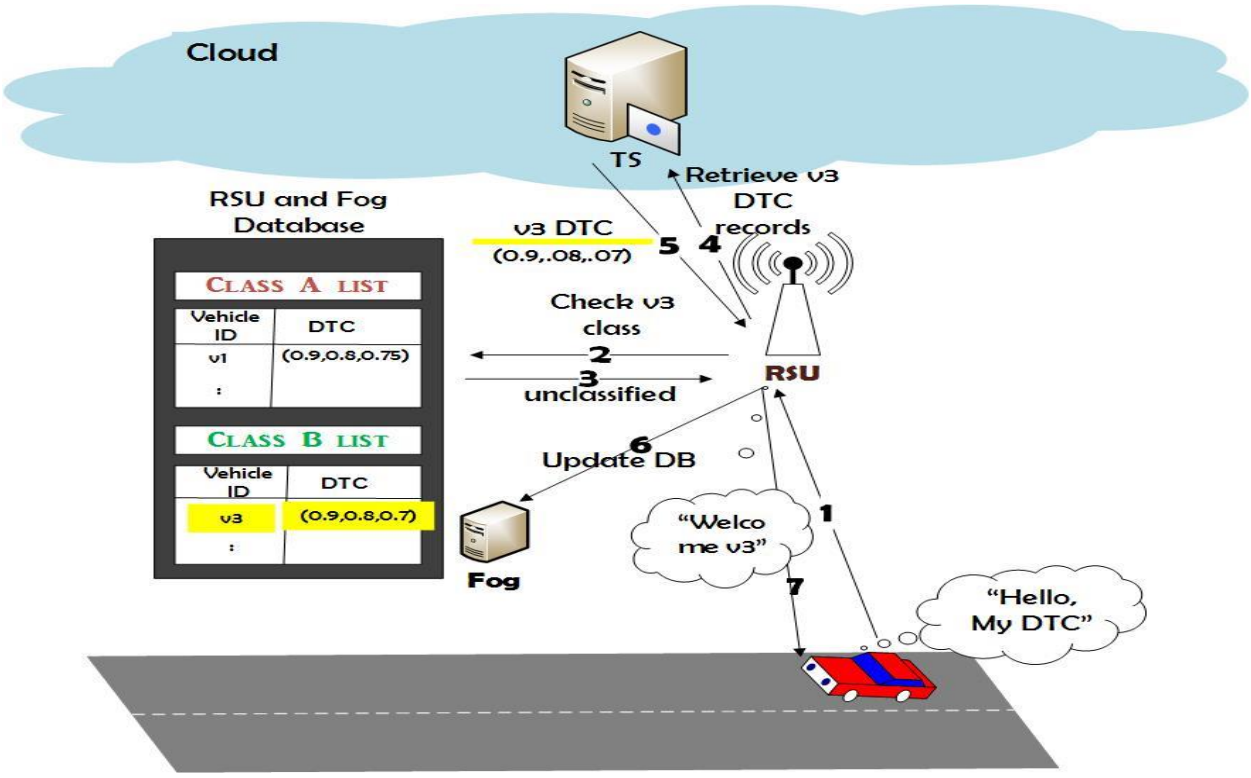


Figure 4.4: Central Retrieval for newcomers vehicles.

ii) ***Central Retrieval*** (for newcomer vehicles)

In contrast to offloaded retrieval, central retrieval is used when an unclassified vehicle passes into an RSU's area. In this situation, the RSU classifies the vehicle as a Class B vehicle and sends a retrieval request containing the vehicle identification number of DTC to the TS. The TS responds with the latest DTC records for the vehicle. When the RSU receives the updated DTC records, it sets a flag for the vehicle in its "Welcome" message. Next, the RSU will update its Class B list and inform its fog nodes about the new Class B member. Note that the RSU can send several vehicles' DTCs in the same request to save communication time. After DTC(s) retrieval, the RSU sends the DTC(s) to its fog nodes (see Figure 4.4). Note that the RSU does not have to send a notification message to the fog nodes in offloaded retrieval because the vehicle's records are already in their database.

***Theorem* 1**: *Our fog-based reputation evaluation model costs the vehicle only one message to request joining the network.*

Proof:

Each vehicle must send a "Hello" message (i.e., joining request) to the RSU with its DTC. Then, when the RSU accepts the joining request, it allows the node to communicate with others. Otherwise, the vehicle is suspended. The flag set in the "Welcome" message determines the state of the node (e.g., fully or partially trusted).

Note that the vehicle can share its TER value when any neighbouring vehicle asks it for advice. For example, if vehicle X seeks advice from vehicle Y (i.e., "Is the road clear?"), vehicle Y replies to X's message with its advice (i.e., "The road is clear"), with its TER value (as an advisor) attached, and with the flag that was earlier sent by the RSU. The flag can be used by the receiving vehicle to detect partially trusted nodes.

***Theorem* 2**: *The workload of the infrastructure (RSU) central retrieval is heavier than the workload of offloaded retrieval.*

Proof:

Offloaded retrieval costs the RSU only ***one*** message to accept/reject the joining request in case of offloaded retrieval. The RSU replies to the "Hello" message by sending a "Welcome" message with a flag that reflects its opinion (i.e., "Accept", or "Reject"). However, in central retrieval, the RSU must send a "Retrieval request" to the TS, then after receiving the reply, the RSU updates its fog nodes with the new list and sends a "Welcome" message to its guest. In short, central retrieval costs the RSU **(*2*+*n*)** messages, where *n* is the number of fog nodes.

***Theorem* 3**: *Evaluation Trust process is more expensive in the experience-based trust evaluation model compared to our fog-based reputation evaluation model.*

Proof:

The node in our fog-based reputation evaluation model relies on its neighbour's reputation (i.e., TER) to establish trust. Thus, no more messages are needed to establish or evaluate trust. The vehicle sends only *one* message (i.e., the "Hello" message) to join the network and after accepting it, can send/answer any message associated with its reputation (i.e., TER) and the flag that is received from the RSU. In contrast, in experience-based trust evaluation models such as [43] and [51], the vehicle must send *N* messages to the vehicles to establish trust (*N* is the number of the neighbouring vehicles). In other words, the number of messages is directly proportional to the number of vehicles in the experience-based trust evaluation model, whereas it is fixed in our reputation model.

4.3 Task-based Experience Reputation (TER)

Trust management in VANETs mainly focuses on evaluating the direct experience between peers without taking into consideration the type of task (i.e., the type of experience). Some tasks affect traffic safety, such as giving misleading advice (e.g., "change to the left lane" and there is an accident in the same lane). On the other hand, some tasks do not affect traffic safety, such as entertainment (i.e., giving a review about a restaurant in the area), but they aggregate with the total reputation value of the vehicle. Thus, it is useful to evaluate each experience based on the type of task that the node performed. Using this

strategy, we can select a specific node to perform a given task (e.g., cluster leadership) based on its TER in performing that type of task.

In this section, we identify the problem of computing the reputation value with disregard to the task type. Also, we show how this problem negatively affects the final trust evaluation value. To the best of our knowledge, we are the first to introduce the "All in the same boat" issue to VANET and suggest classifying the experience between vehicles based on the task type.

➢ "All in the same boat" of updating trust and reputation values in VANETs

Most reputation models (i.e., entity-based and hybrid models) mainly depend on the history of the nodes, which is a significant factor in evaluating the trustworthiness of the node. The decision of accepting or rejecting the node's advice or relying on that node is based on this evaluation. The experience-based trust model, where the real-time experience is evaluated, could be inaccurate due to the ephemeral nature of VANET. In other words, the vehicle may not have the chance to interact more than once with its neighbour because of the highly dynamic nature of VANETs. Not only that, the model may fail in the situation where the node cannot perform even one interaction due to the high mobility or the existence of obstacles (e.g., buildings). Therefore, recommendation-based models have been proposed to support the scenarios where the node fails to gain experience-based trust. In these models, the node builds its trust according to its neighbours' recommendations.

Simply, the node can trust or distrust its neighbour based on the evaluations of the other neighbours if these neighbours (i.e., recommenders) are trusted by the node. In this framework, the node does not have to directly interact with its neighbour to evaluate trust since it can get it indirectly from the received recommendations. In a few words, the real-time experience and the history of nodes are foundation stones in trust-based (direct real-time experience)/recommendation-based (indirect real-time experience) and reputation-based (previous experiences) models, respectively. However, updating trust or reputation values merges all experience evaluations into one value, which gives a low-resolution evaluation that may be inaccurate.

To explain the problem of the "All in the same boat", we consider the following situation: there are two vehicles (v_1 and v_2), and two tasks (s_1 and s_2) required from both nodes. Task s_1 is reporting a major accident on the road, and task s_2 is providing recommendations without being requested. Let us assume that the effects of both tasks (call them *E1* and *E2*) are different. Task s_1 has a higher effect than Task s_2 because task s_1 is a safety-related task and task s_2 is a non-safety-related task (i.e., $E1 > E2$). If the node performs a task well, its trust evaluation T would be 1(i.e., trusted); otherwise, T would be 0 (i.e., distrusted). The final trust (T_{avg}) is computed. The trust threshold is 0.7, and the effect of both tasks is ignored.

Suppose that v_1 performs well in task s_1 and badly in task s_2 and further suppose that node v_2 performs the opposite (badly in task s_1 and well in task s_2)**.** The R_{avg} of both v_1 and v_2 are the same. However, node v_2 caused a damaging effect (e.g., loss of

lives) when it performed the safety-related task badly, whereas the negative effects from node v_l are minor (plus, v_l contributed positively when tasked with the safety-related task).

Mixing the evaluations of different types of experiences in one aggregated value increases the risk of leading to inaccurate evaluation (all experiences in one value). The currently used methods of aggregating the evaluations are mostly unfair because it gives similar weight to all types of experience without considering the impact size of each experience.

➢ TER calculation

In this section, we elaborate on how our proposed solution can prevent the selfish/malicious nodes from increasing their reputation values where they cause damage in VANETs. As we mentioned earlier that reputation updating in VANETs may suffer because of the "All in the same boat" issue. We highlighted two common reputation updating methods that are also used to update trust values, as follows:

Method 1: *Aggregated Reputation* in [82], where the reputation is updated by taking the average of the current reputation values and the aggregated reputation is defined as:

$$Rep_{new} = \frac{1}{N}\sum_{k=1}^{N} Rep_{previous}$$

where $Rep_{new} \in [0,1]$ represents the updating reputation value given to a vehicle, N is the number of the reputation values, and $Rep_{previous}$ denotes the previous reputation values and $Rep_{previous} \in [0,1]$.

Method 2: *Accumulated Reputation* in [51], where the reward and punishment technique is applied. The vehicle with reliable advice is rewarded (i.e., its reputation value is increased), and the vehicle with unreliable advice is punished (i.e., its reputation value is decreased). The forgetting factor does not integrate into this method, which gives the same weight for all experiences regardless of their freshness. However, this method is valuable to motivate nodes to build up trust and get rewards.

The basic idea of our proposed solution is computing what we called it, Task-based Experience Reputation (TER), where the reputation value of each task type is computed separately (i.e., multiple TER values). This simple solution can guarantee getting an accurate evaluation that reflects the node's performance in the past, depending on the task type. Moreover, it reveals the weakness and strength of the node in doing a certain type of task. The solution mainly does not mix up all the node's experience evaluations in one value to avoid the malicious nodes taking advantage of their good attitudes to cover their misbehaving attitude. In addition, this solution can be used with any updating methods; however, the key secret is separating the evaluation of each type of task. For example, the node can have TER_1 and TER_2, where the first value presents its reputation value of safety-related tasks, and the second value presents its reputation value of non-safety-related tasks.

➢ TER and Smart Employment in VANETs

By applying TER, we can improve the evaluation process. Furthermore, we believe that if the method of evaluating trust could reflect the performance level according to specific criteria as well, this will be a significant step towards smart employment of vehicles on the road. The infrastructures and vehicles can employ/ rely on not only the most trusted node but also the most competent one at performing a specific type of task.

4.4 Performance Evaluation

In this section, we set a series of scenarios and analyses to demonstrate the performance of our fog-based reputation model. First, we compare the performance of the proposed model with the experience-based trust models in [8] in terms of message transmission overhead and the workload of the vehicles and infrastructures. Then, we validate the variations of reputation values of two types of updating trust evaluation models in [51,82]. We shed the light on the problem of both updating models. Finally, we applied our solution and showed how it improves the reputation updating schemes.

➢ Message Transmission Overhead

In this part, we validate the number of required messages that the node must send in the experience-based trust models [8,51] and our model. In this scenario, about 200 vehicles are randomly distributed in the network, and they are in the same geographical area where there is only one RSU and three fog nodes. In the event detection models in [8, 51], the experienced-based trust is used to confirm the existence of the event, the nodes mainly rely on the real-time experience to evaluate trust. Thus, each node must interact with its neighbours to evaluate their trust. However, in our proposed model, the node must send a “Hello” message to the RSU, and then the RSU directly responds with a “Welcome” message to the node, and the node does not have to communicate with any vehicles for any evaluation purposes.

We assume that the vehicle can communicate between10% up to 50% of the total number of vehicles in the network. Figure 4.5 shows the total number of messages sent around the network in the experienced-based trust models. The number of messages is increased according to the number of the vehicle's neighbours that the nodes have to deal with. Note that the node cannot interact with all nodes in the network due to the limited transmission range and the speed of the vehicles; thus, the size of communication (the percentage of nodes) above 30% is considered unrealistic in our scenario.

As shown in Figure 4.5, when the number of neighbours of each vehicle is about 20, which is 10% of the total vehicles, the number of messages that are needed for evaluation purposes is about 7800 messages. This number of messages includes the forth and back messages. By increasing the number of neighbours that the vehicle connected with, up to 100 vehicles, the number of messages is increased as well to be close to 40 thousand messages. In our proposed model, the number of messages

is varied according to the number of well-known nodes. Also, in our proposed model, the nodes have to deal only with the RSU only to verify their trustworthiness. Figure 4.6 shows the number of required messages exchanged between the nodes, the RSU and the TS. In the worst-case scenario, where all vehicles are newcomers, the number of messages increases to 1400 messages with no well-known vehicles. Our proposed model has a lower transmission message overhead than the experience-based trust model because the infrastructure only is responsible for validating the node's trust.

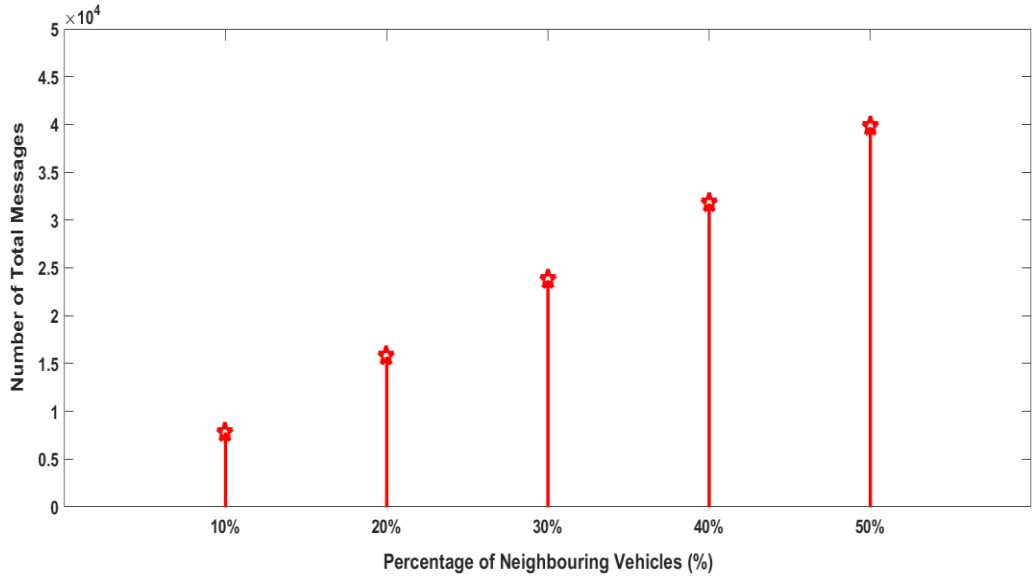


Figure 4.5: Messages overhead in the experience-based trust model.

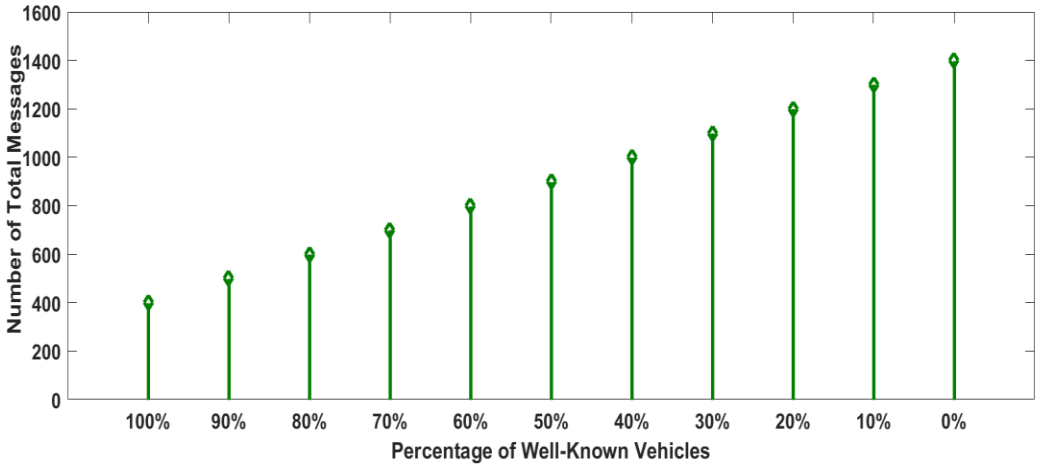


Figure 4.6: Messages overhead in the Fog-based reputation model.

➢ The workload

As shown in Figure 4.7, the workload of the trust evaluating phase in the experience-based trust models (models that depend on the real-time experience) is 100% on the vehicle, which is responsible for interacting with its neighbours and evaluating and updating their trust. However, in our fog-based reputation model, the vehicle does not have to interact with its surrounding vehicles for evaluation purposes because the reputation of the vehicle is used instead of the real-time experience trust. Therefore, to allow the vehicle to communicate and share its reputation values (TER) with others, the vehicle only has to send only a single message (i.e., "Hello" message") to the RSU to accept its joining request or refuse it (blocking it from communication fully or partially). Our proposed model reduces the workload of the vehicle from 100% to 50% down to 17% because the infrastructure (the RSU and the TS) is responsible for trust evaluation instead of the vehicle. In the offloaded retrieval, the RSU has a similar workload of the vehicle (Case 1). In other words, the vehicle sends a "Hello" message and RSU replies by a "Welcome" message. However, in the central retrieval (Case 2), the RSU has to communicate with TS to retrieve the vehicle's DTC and then it sends its "Welcome" message and notifies its fog nodes about the new member (we assume that we have three fog nodes are under the supervision of RSU).

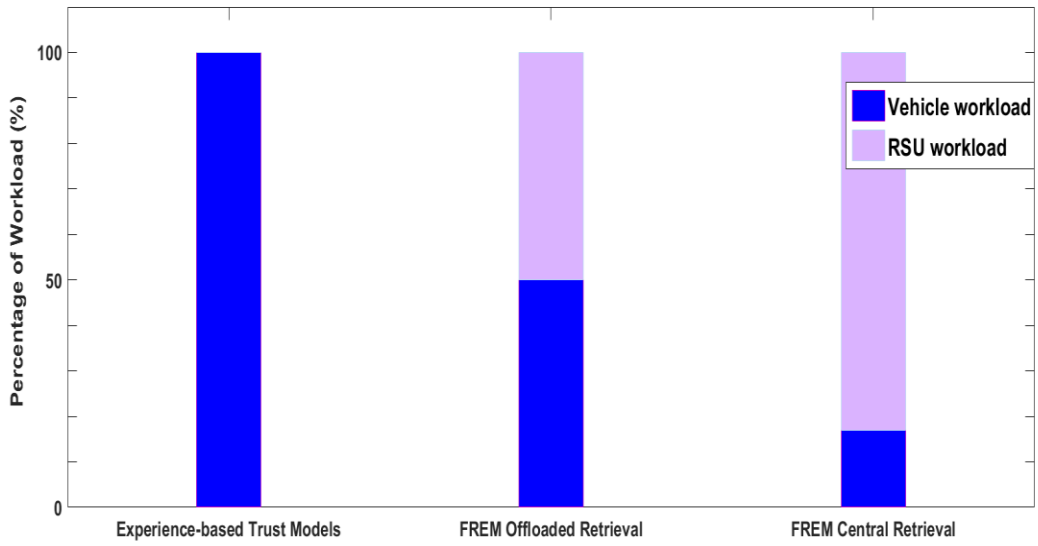


Figure 4.7: The workload of both the RSU and vehicle in the existing trust and FREM model.

➢ The evaluation of the existing trust updating methods

First of all, we would like to show the effect "All in the same boat" issue in both methods. We show the reputation value of six nodes with different misbehaviours. We give each node two different types of tasks (i.e., safety-related and non-safety-related tasks). The number of tasks in both types is similar (i.e., 50% are safety-related tasks and 50% are non-safety-related tasks). The probability of honestly performing the tasks for node 1,2,3,4,5, and 6 is 100%,90%,80%,70%,60%, and 50%, respectively. In other words, nodes 1, 2,3,4,5, and 6 failed to do the required safety task by 0%,20%,40%,60%,80%, and 100%, respectively. The nodes only misbehave in the safety-related tasks where they are doing well in the non-safety-related tasks. We assume two different scenarios based on the nature of each task. The performance of the node in the required tasks in scenario 1 is evaluated based on how much the vehicle (receiver) is satisfied with the sender's performance. On the contrary, the range of values of all personal experienced-based trust can be set to either 0 or 1in scenario 2. In other words, in scenario 1, the nature of this task requires the evaluation to be in the interval of (0,1) whereas, in scenario 2, the evaluation of the task is binary, either 0 or 1, where 0 is for bad performance and 1 for good performance.

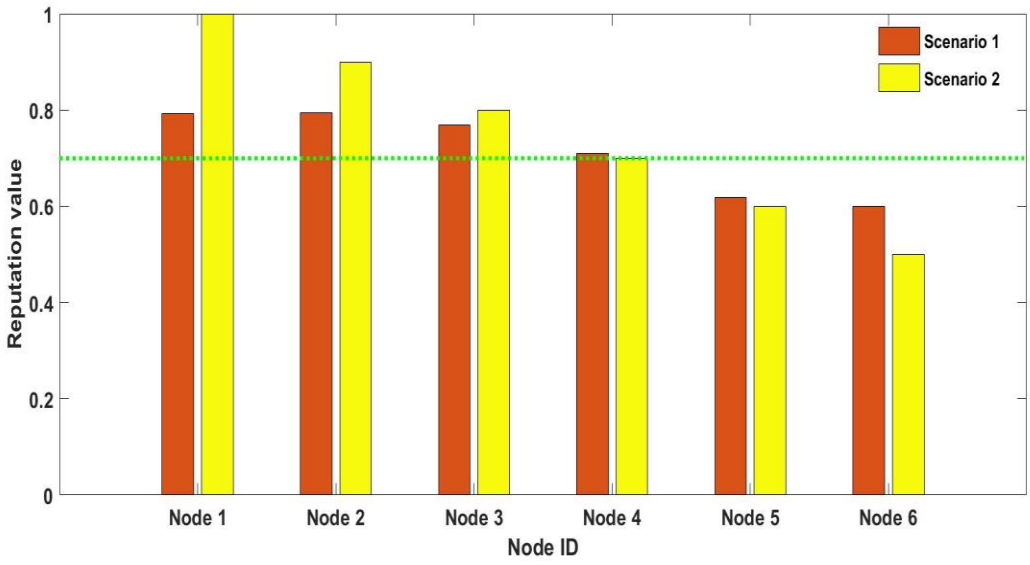
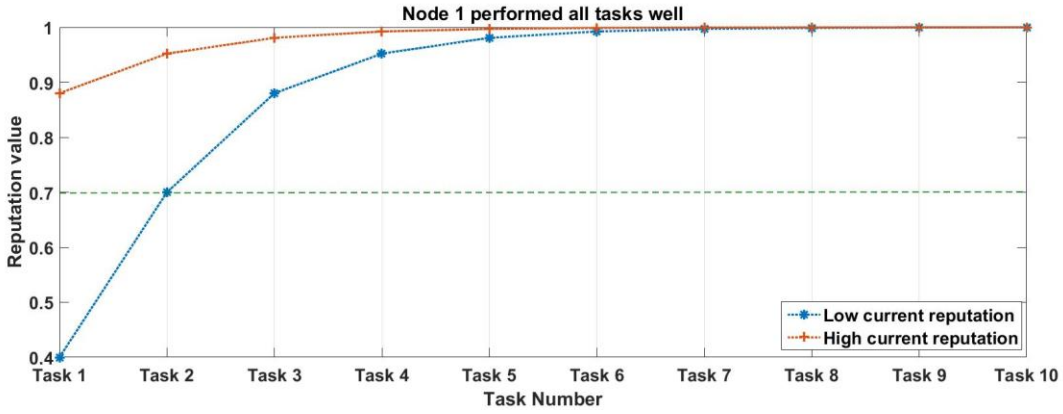


Figure 4.8: The reputation value of nodes using Aggregated Reputation.

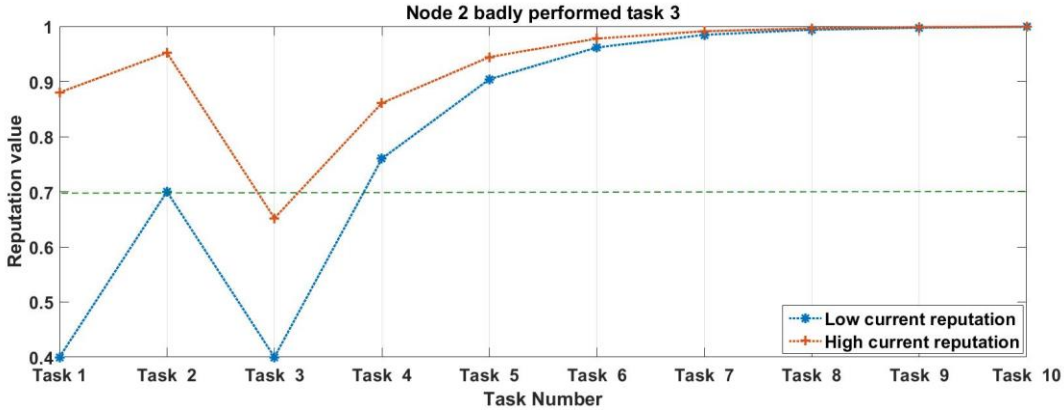
In the first method, we verify the performance of the updating model in [82]. In Figure 4.8, the red bar presents the first scenario, and the yellow bar presents the second scenario. The green dashed line is the reputation threshold (i.e., 0.70). Figure 4.8 plots the performance of each node in both scenarios. As shown in Figure 4.8, the probability of successfully performing the given tasks of a node is decreased, which is reflected on the reputation value of each node. It can be seen that scenario 2 is more generous with nodes 1, and 2 (i.e., it gives high reputation values to the nodes with high performance) compared to

scenario 1 because the number of successfully performed tasks is still high. The performance of node 1 is the highest performance about 80% and 100% in scenarios 1, and 2, respectively. Node 1 can honestly perform all tasks. Comparing all reputation values for all nodes under both scenarios, we observed that even when node 4 failed in about 60% of the safety-related tasks, it is still trusted, and other neighbours can rely on it to do more tasks. In other words, the node can gain an acceptable reputation value even it negatively affects the network by its horrible performance in safety-related tasks. The reputation value of node 1 in scenario 2 is higher by 20% than in scenario 1, which indicates that the nature of the task (i.e., the evaluation mechanism) may affect the reputation value.

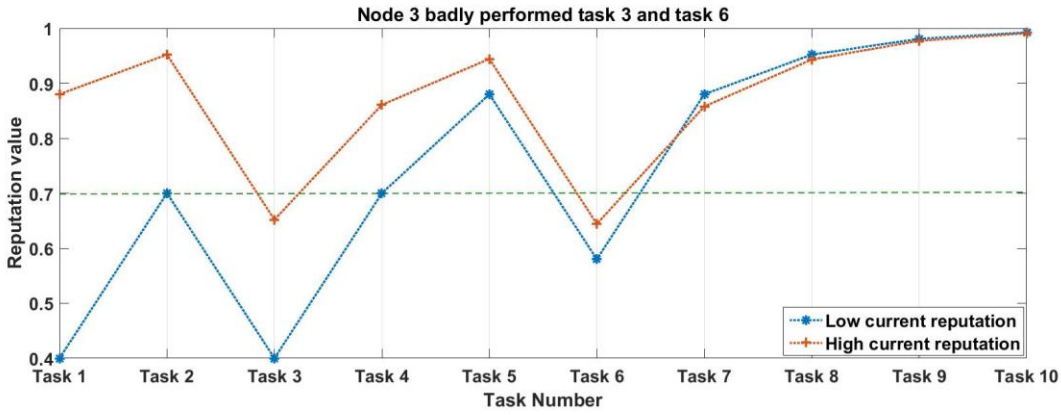
In the second method, we verify the performance of the trust updating model in [51]. This method is also a commonly used method to update reputation and trust values in VANETs [51]. We validate the reputation value of previously mentioned nodes, see Figure 4.9. We present the behaviour of the nodes based on two different current reputation values (i.e., low and high reputation values). The blue dashed line and the red dashed line reflect the reputation value at each time the node is done from a given task. We use 0.1 as a low reputation value and 0.7 as a high reputation value of the current experience (i.e., Task 0), respectively. The green dashed line is the trust threshold which is 0.7. The six nodes have to do the same tasks (10 tasks). The reward and punishment factors are set to be 0.6, and - 0.6, respectively. The number of both types of tasks (the safety-related and non-safety-related tasks) is similar (i.e., 5 safety tasks and 5 non-safety tasks). Figure 4.9 (a) shows Node 1, which perfectly performs all the required tasks. We realize the small increase in the reputation value of node 1 at task 1(from 0.1 to 0.4) but because of the low current reputation, the node is defined as distrusted at this point. Nodes 2,3,4,5, and 6 show poor performance in doing the safety tasks only. Node 1 badly performs one task (i.e., 1 out of 5 safety tasks) and nodes 3,4,5, and 6 fail to perform 2,3,4, and 5 safety-related tasks, respectively.



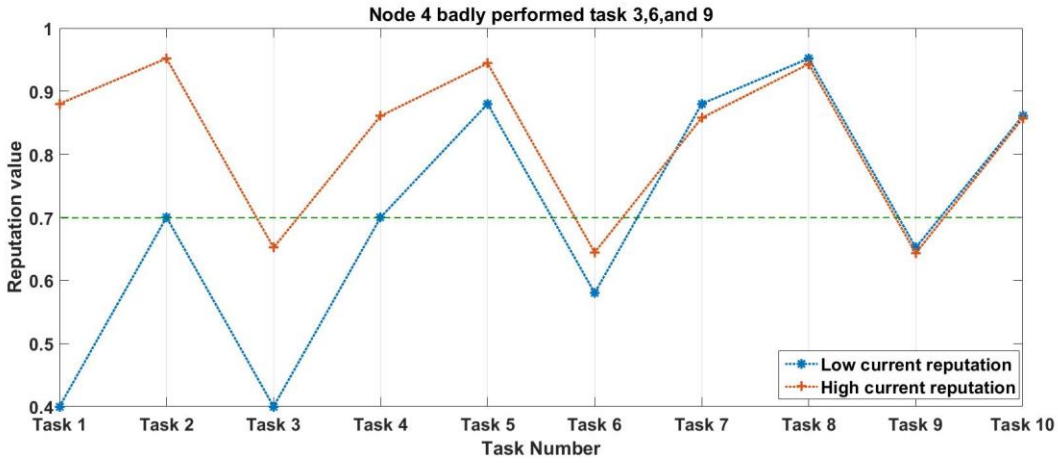
(a)


Node 2 badly performed task 3
Reputation value
Task Number
Low current reputation
High current reputation
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10

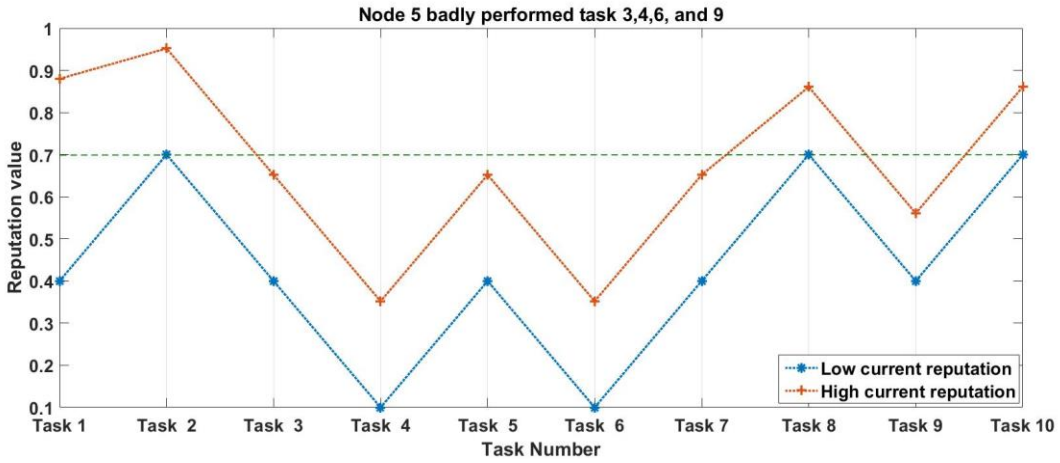
(b)


Node 3 badly performed task 3 and task 6
Reputation value
Task Number
Low current reputation
High current reputation
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10

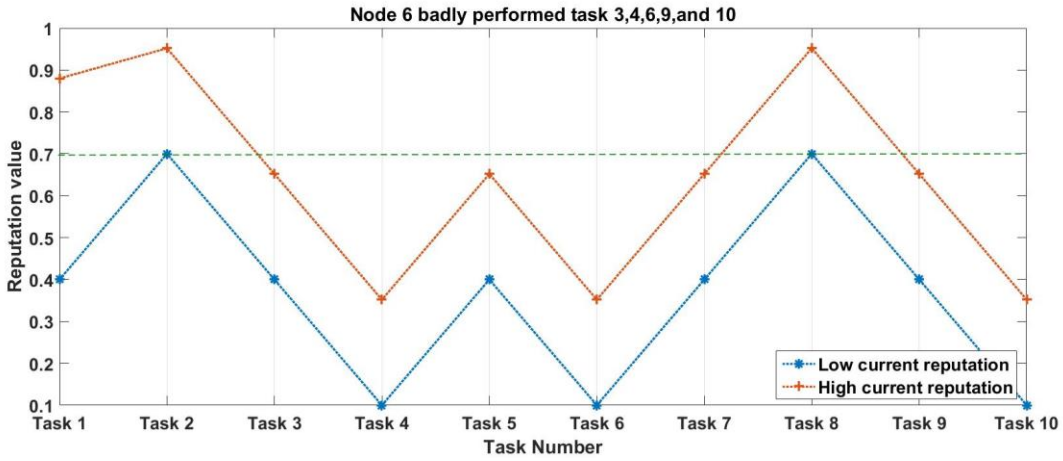
(c)


Node 4 badly performed task 3,6,and 9
Reputation value
Task Number
Low current reputation
High current reputation
Task 1
Task 2
Task 3
Task 4
Task 5
Task 6
Task 7
Task 8
Task 9
Task 10

(d)



(e)



(f)

Figure 4.9: The updated reputation values of all nodes task by task using Accumulated Reputation.

From Figure 4.9, we realize that this method quickly responds to the node's behaviour and reflects the change of the node's behaviour on its reputation value. The final reputation values of each node after doing the ten tasks are shown in Figure 4.10. We also reveal the earlier mentioned issue (i.e., "All in the same boat") still appears in this model; however, it is clearer here as we will see, and each node is incorrectly defined as trusted where it is not. All nodes with different behaviours are assigned as trusted nodes except node 6. Thus, method 2 is prone to trick by the malicious nodes that change their behaviour periodically, which allows them to get more rewards that increase their reputation values.

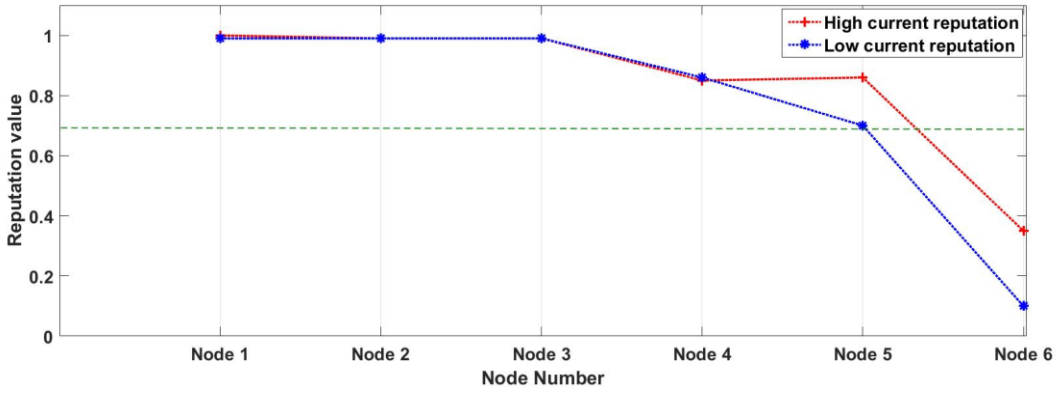


Figure 4.10:The final reputation value of nodes using Accumulated Reputation.

➢ The effectiveness of using TER

In this subsection, we applied the TER concept on the aggregated and accumulated reputation methods (i.e., computing the reputation value of each node in the given safety-related tasks only). Figure 4.11 and Figure 4.12 illustrate TER for all nodes in both scenarios 1 and 2, respectively. The red dashed line shows the trust threshold. We have the following observations of these two figures: a) node 1 and 2 with 100% and 80% good performance are assigned as trusted nodes, b) the other nodes are clearly defined as distrusted nodes, and c) regardless of the way of evaluation (Boolean or not), the nodes in both scenarios (with different nature of tasks) are similarly defined) trusted and distrusted).

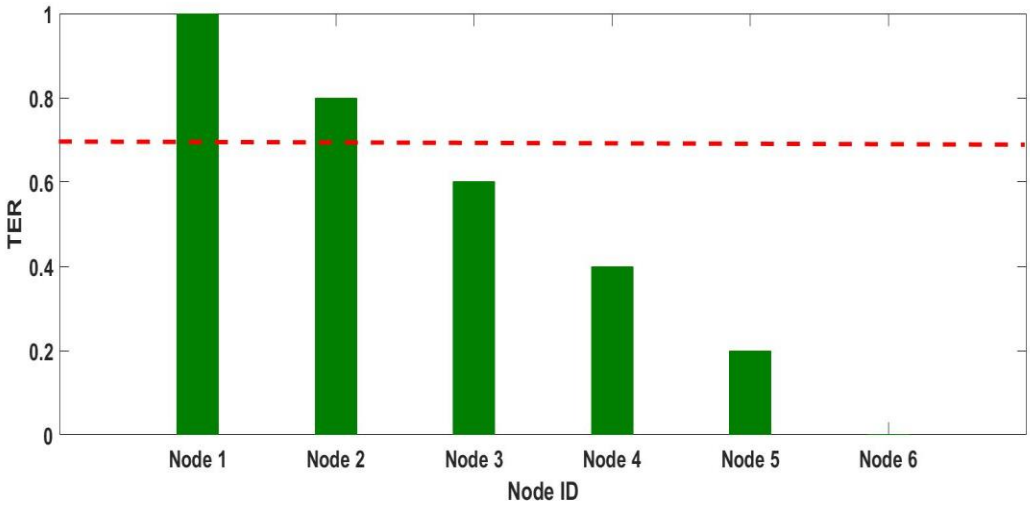


Figure 4.11: The (TER) value using the proposed solution of Aggregated Reputation, Scenario 1.

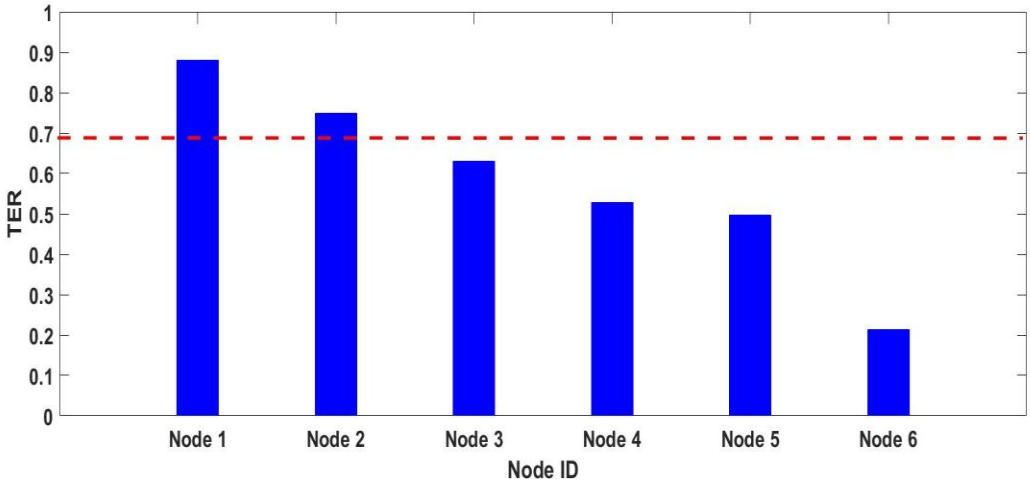


Figure 4.12: The (TER) value using the proposed solution of Aggregated Reputation, Scenario 2.

Moving to method 2, where we assume that the nodes start with low and high current reputation. Figure 4.13 shows the same results as Figure 4.11 and Figure 4.12. We observe that the starting current reputation in Accumulated Reputation, either low or high, it does not affect the final evaluation in this scenario. Also, TERs are high for only node 1 and node 2 because they do well in their given tasks. Nodes 3 failed to get high TER, whereas nodes 4,5, and 6 are shown as absolute distrusted nodes due to their misbehaving attitude (i.e., they got '0' reputation value).

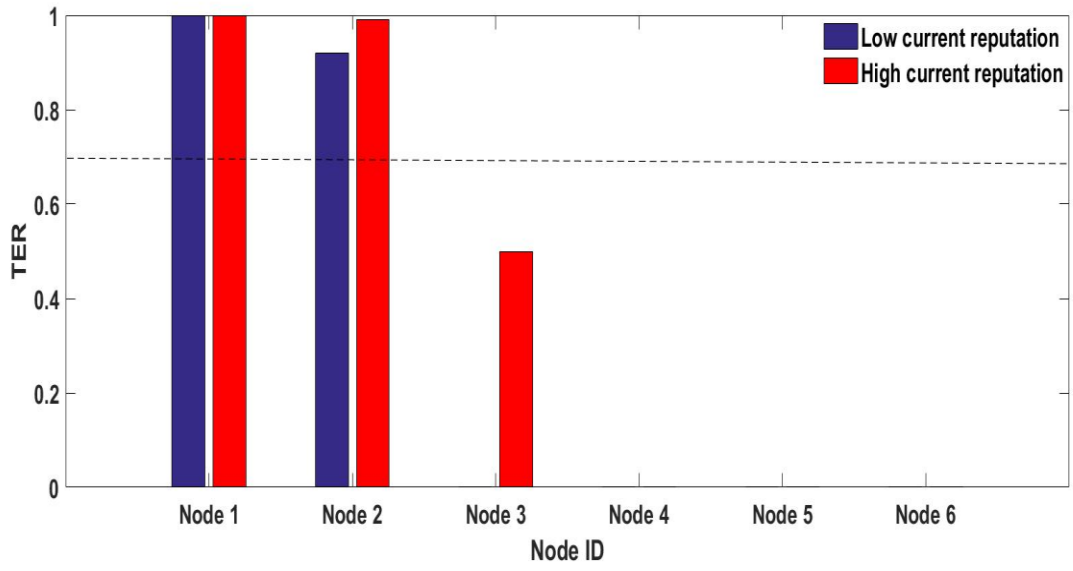


Figure 4.13: The (TER) value using the proposed solution of Accumulated Reputation.

Table 4.1 concludes the evaluation of each node according to different methods of reputation updating. We investigate that using aggregated and accumulated reputation methods for updating the reputation or trust is inaccurate because the malicious

node can use its performance in non-safety-related tasks as a cover to pass its unlawful operations, whereas separating the reputation value of the tasks according to its type perfectly reflects the node's reputation in doing such a task. Based on that, our proposed model increases the credibility of the reputation evaluation system. In addition, it can support smart employment in VANET. In other words, the infrastructures or the vehicles can rely on the TER values of the node to assure the right level of the node at performing a specific task, which will be more accurate than using a general reputation value that reflects the history of the nodes in different tasks. According to the results in Table 4.1, we can rely on node 1 and 2 to perform any safety-related task; however, by following aggregated and accumulated reputations, although nodes 3, 4, and 5 showed a bad performance at doing safety tasks, the models of 1 and 2 still choose them. Thus, methods 1 and 2 may take the wrong decision to rely on nodes 3, 4, or 5 (distrusted nodes). From the above scenarios, method 1 picks two wrong nodes (i.e., node 3 and 4) and defines them as trusted, whereas they are not. The same thing happens with method 2, where nodes 3, 4, and 5 are defined as trusted, yet they are distrusted nodes. Therefore, by improperly defining the node, we will give a chance to a malicious node to affect the network and the driver's safety.

Table 4.1: The conclusion of Aggregated and Accumulated Reputation and their proposed solutions.

Method Name	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
Aggregated Reputation	Trusted	Trusted	Trusted	Trusted	Distrusted	Distrusted
Accumulated Reputation	Trusted	Trusted	Trusted	Trusted	Trusted	Distrusted
TER for Aggregated Reputation	Trusted	Trusted	Distrusted	Distrusted	Distrusted	Distrusted
TER for Accumulated Reputation	Trusted	Trusted	Distrusted	Distrusted	Distrusted	Distrusted

4.5 Discussion

- *Reputation Aggregation*

The main function of fog nodes located in the same area is aggregating DTC records of the road users (i.e., vehicles on the road). In a clustered network, each vehicle must send its feedback about vehicles that interact with to the CH, and then the CH sends the feedback to the nearest fog node before ending the session or after CH death (i.e., no members on its cluster). However, in non-clustered networks, the vehicles have to send feedback directly to the nearest fog node. Thus, fog nodes in the same area may receive several instances of the same information from a vehicle (i.e., duplicate instances). There are three possible solutions to overcome the duplicate instance problem: 1) fog nodes send all the collected data to the RSU, and the RSU ignores duplicated instances, 2) a vehicle cannot send the same instance twice (i.e., network condition), or 3) one fog

node can be assigned (e.g., master fog node) to make sure there are no duplicate instances of data before sending data to the RSU. Earlier in this chapter, we introduced the third solution, where fog nodes may collaborate to dynamically assign this task to the fog node that has the least workload at the moment of sending the data records. Note that fog nodes must incorporate an effective algorithm to eliminate the malicious reputation (i.e., badmouthing attacks) besides eliminating the duplicate instances. According to [84], fog can compute the trust score of the vehicle on-the-fly without interacting cloud too. That means that fog can be an evaluator by itself rather than a records collector. Therefore, fog in our context can construct their own reputation system (local systems) since they can identify their local people.

- *Traffic Mobility and DTCs Retrieval*

Traffic mobility is determined based on the location (e.g., highway or urban) and time (e.g., rush hour or ideal hour) of the area. In other words, different areas may have lower or higher traffic mobility at different times during the day (e.g., highways have higher traffic mobility than urban areas at rush hour). DTC retrieval is not considered a challenge with respect to class A/B vehicles because fog can use offloaded retrieval without the need to communicate with the TS. However, for newcomers vehicles, the RSU has must send a request to the TS to retrieve DTCs. This is not a big issue in urban areas where traffic mobility and the probability of having a new visiting vehicle is lower compared to areas with high traffic mobility, such as highways; however, in the areas with high mobility such as highways, the challenge of devising an algorithm that responds quickly to DTC central retrieval is increased.

- *Competency and Trust*

We evaluate the qualification of a node to do a certain task based on the metric of trust. More metrics can be proposed to evaluate the qualification of the node, such as the sensitivity, and the complexity of the task, how many times the nodes successfully perform this task compared to the total number of tries, the node's flexibility, meeting the network requirements, node's fast response, and the network topology complexity level. Also, some questions arise with the introduction of competency in our model, such as Should we think in terms of competency when selecting a node for a task on VANET instead of focusing on the trust metric only? How can we evaluate the competency?, What happens if a node has the highest competence rating but a slightly lower trust evaluation?, and Can competency stand alone in a VANET environment without trust, or must they be combined?

4.6 Conclusion

Existing trust models evaluate direct trust irrespective of the task type and delete the value from the vehicle's memory immediately after taking action. Consequently, we cannot develop in-depth knowledge about the vehicle's behaviour. Moreover, we cannot use this direct trust value later. Therefore, in this chapter, we have proposed a fog-based reputation evaluation model for VANETs, it has the following: 1) the proposed model integrates fog in VANET and takes advantage of its capabilities (i.e., storage and computing), which reduces the cost of deploying other expensive infrastructure such as RSUs. Fog nodes are used as DTC record keepers in the long-term or short-term, 2) the message complexity of the trust establishment phase is reduced by using DTCs, 3) the proposed model recognizes that vehicles have preferred paths that they travel daily or weekly. The model is designed so that fog knows more about the most visited vehicles. Therefore, employing the node who is both trusted and an expert for a given task can be achieved (i.e., the trust metric may not be enough but experience in such a task is required), and 4) The proposed model may perform well in areas with low traffic mobility, such as urban areas. Also, we illustrate some challenges when it comes to high traffic mobility areas, such as highways may leave for future work.

Chapter 5:
FEVM: Fog-based Event Validation Model

5.1 Introduction

A high-level challenge for VANETs is to extract an accurate understanding of reality from the plethora of information transmitted throughout the network. Regarding an event's occurrence, information received from different nodes may express different levels of confidence or divergent opinions on the circumstance. Further, the information may be accurate, inaccurate, or fraudulent. VANETs must aggregate this spectrum of information in a manner that seeks to optimize the success rate of the event validation process, and hence the decision-making process. A goal of FREM in Chapter 4 was to expand the role of fog in evaluating an event's state, and the Fog-based Event Validation Model proposed in this chapter extends this theme. Fog is tasked in this model with decision-making based on the available information drawn from its nodes (i.e., the vehicles). To this end, fog capitalizes on the mobility of vehicles, which permits them to gather data from the network through their trips. From this perspective, the vehicles in FEVM act as mobile fog nodes that support fog's event validation process, which is handled by the fixed fog nodes.

As discussed hitherto, the levels of confidence in an event held by a given vehicle, the levels of trust extended to other nodes sharing information, the historical reputation scores of nodes, etc., are affected by numerous factors, including experience, vehicle role, sensor technology, etc. In Chapter 4, we introduced TER, adding yet another dimension along which we ascribe varying reputation values. Thus, when presented with reports from different vehicles, each possessing different confidence and reputation values, it is natural to place different levels of importance on each when aggregating the information they convey. In [86], a method for combining evidence using different importance factors was proposed as an extension to the Dempster-Shafer Theory of Evidence (DSE) [86] for validating events in Mobile Ad-hoc Networks (MANET). The Dempster-Shafer Theory of Evidence (DSE) provides a method for representing uncertainty, and for combining multiple points of evidence [87]. Here, we apply the method evolved in [86] to the event validation process of the VANET. Assigning the task of aggregating evidence (using importance factors) to fog aligns with our stated objective of expanding the role of fog, while alleviating the computational burden on vehicle nodes. Vehicles, on the other hand, aggregate messages from peers using a more conventional strategy based on the method of trust aggregation proposed in [88]. The method in [88], which incorporates the roles and experiential trust of peers, is extended here to incorporate the number of reports, relay hops, and reputation. Reports sent from

vehicles to fog are aggregated along with all available information to estimate the probability of an event. The resultant probability estimate forms the basis for making decisions in the VANET.

5.2 Proposed Model

In this section, we describe the proposed Fog-based Event Validation Model (FEVM) in detail. Fog nodes, which are deployed along the roads and connected through secure links, play an essential role in the FEVM. One of our main design goals is to emphasize fog involvement by appointing fog nodes with decision-making authority for all vehicles in their respective zones. A key benefit of this design choice is that the decision-making process in a given zone is unified since the fog nodes stipulate one decision (on an event's occurrence) on behalf of all vehicles in their zone. For the purpose of event validation, fog collects data from vehicle reports and from neighbouring IoT devices (e.g., cameras, sensors, and radars) installed in the same zone to accumulate as extensive and reliable information about the current road conditions (e.g., traffic congestion) as possible. There is no crossover of communications from IoT devices to fog and from vehicles to fog; therefore, data received from these two sources is independent. This has two main advantages: 1) the speed and resolution of event detection are elevated since IoT may detect an event before any vehicle, or vice versa, 2) independent sources of information provide greater reliability in determining the true event state (e.g., it is more difficult to corrupt independent sources). The synergistic combination of vehicle and IoT nodes, therefore, permits fog to extract a more reliable interpretation of reality. Strategic positioning of the fog nodes further equips them to act as the middleman, exchanging information between the RSU in the intermediate layer and vehicles or IoT devices in the edge layer.

5.2.1 FEVM Topography

Neighbouring fog servers cooperate, forming a virtual chain along a transportation route served by a RSU. Each fog server can be viewed as a link in a chain having jurisdiction over a specific zone within the route. The zones assigned to the fog servers are contiguous; hence all points along the transportation route lie within a zone, and each zone is governed by a single fog server. For example, consider the transportation route illustrated in Figure 5.1, comprising three zones, each overseen by its respective fog server. Vehicles enter at Zone 1 and travel towards Zone 3. Vehicles entering the transportation route must send network joining requests to the RSU to receive the services of the network. Once the network joining request is received, the RSU verifies the vehicle credentials (e.g., ensure the vehicle has a validated trustworthiness card DTC). If the criteria are met, the RSU authorizes the network joining request and produces an estimated timestamp for the vehicle's incorporation into the network. As vehicles travel through the zones, they are subject to decisions made by the fog server overseeing the zone they are located.

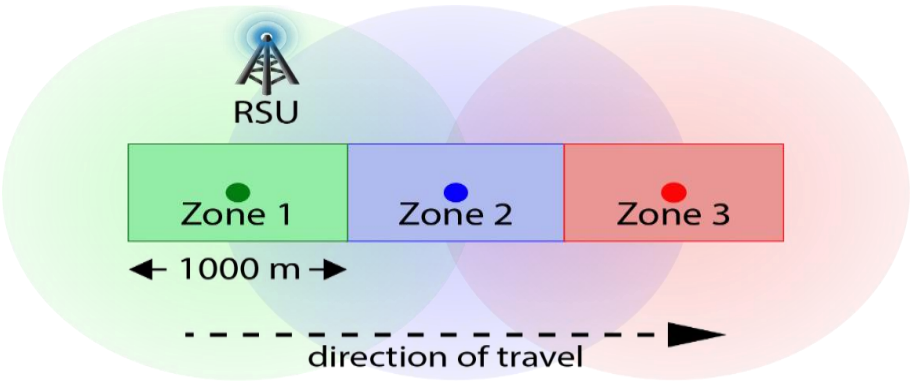


Figure 5.1: Transportation route with RSU and fog servers serving three zones.

5.2.2 Event Evaluation and Validation

Suppose at time t_E an accident occurs in Zone 3. Vehicles directly involved in the accident automatically generate event reports based on the vehicles' onboard sensors. Vehicles within the short range of the event may also generate reports, relying on those vehicles' sensors and drivers to report the event (automatically or manually). All generated reports are immediately transmitted to vehicles within range and to Fog 3. Fog 3 relays all reports to Fog 2, and Fog 2 relays all reports from Fog 3 and its vehicles to Fog 1. Vehicles receiving reports will aggregate the reports after a threshold is met (either a minimum number of reports or an elapsed time, $T_{threshold}$, since the event start time, E_{st}) to arrive at a confidence level for the event and then relay this information to other vehicles and the local fog server. Similarly, after meeting one of the threshold criteria, Fog 1 aggregates the report information received to estimate the probability the event is true and make a decision. The decision is immediately promulgated to all vehicles in Zone 1. In other words, if Fog 1 decides the event exists, it warns all vehicles in its zone (i.e., the currently available vehicles and the newcomer ones). Also, it forwards its decision to Fog 2.

Note that if Fog 1, Fog 2, or Fog 3 receives confirmed information from IoT device(s), they make a decision without having received reports from vehicles in their zones. Otherwise, each fog must hear from its vehicles and from other fog nodes to make its decision. Also, it is important to note that fog servers rely on each other's decisions since all are assumed to be trusted and serve the same transportation route.

5.2.3 FEVM Assumptions

- **Transportation Route Topography**

  Each transportation route comprises one RSU, *n* zones, and *n* fog servers. The fog servers are in the transmission range of neighbouring fog servers such that information from any fog server can reach all other fog servers along the route (i.e., Fog A can communicate with Fog B, Fog B with Fog C, and so on). The fog servers are, therefore, sequentially linked to form a virtual chain of contiguous fog zones partitioning the transportation route.

- **Network Membership**

  All vehicles entering the transportation route are associated with a DTC (as defined in Section 4.2), which contains the vehicles' TER values and is stored on a Trust Server (TS). All vehicles entering the transportation route send joining requests to the RSU, which are accepted or rejected by RSU based on network-defined credentials, including the DTC validity and the vehicle's trustworthiness. RSU periodically sends the list of active vehicles in the transportation route to fog servers with a prediction of the route exit time for each vehicle.

  There are two types of vehicles:1) High Role, HR vehicles (e.g., governmental vehicles) comprise 10% of the VANET membership, and 2) Low Role, LR vehicles (e.g., regular vehicles) with LR vehicles making up the remainder. All government and public transportation vehicles, fog nodes, and IoT devices are assumed to be honest.

- **Event Detection**

  All vehicles within 50 m from an event can detect or verify the event by means of the vehicle sensors or the drivers' biological senses. Each vehicle's detection capability, quantified with ξ, is predetermined and is not calculated in this work. Upon generating an event report, the generating vehicle's location and the reported event location are measured/estimated by the vehicle's sensors and recorded on the event report. All events are categorized as Major or Minor, and the simulated event duration is automatically set based on the category (e.g., Major events are automatically set to durations of 60 min).

- **Event Validation**

  A vehicle must either detect an event or receive at least one event report from a peer in order to participate in event validation. All vehicles involved in the route are categorized as event report *generators* (vehicles directly involved in the event or that can detect the event) or event report *propagators* (vehicles that do not meet the *generator* criteria but have received and relayed event reports). Reports received during the event validation process may be consistent or contradictory concerning an event's occurrence. As stipulated in Section 4.2, fog servers possess updated records of TER values of the vehicles in their zones, and these values are relied on to calculate event confidence levels (direct experience

trust scores are not incorporated). Only fog makes decisions regarding the occurrence of an event and is therefore responsible for updating vehicles with road conditions and event notifications following aggregation of evidence.

- **Security**

  All communications throughout the VANET are secured. Privacy issues, which have been the focus of numerous studies, are outside the scope of this work. No vehicle can manipulate its reputation value, and this value may be accessed by any other vehicle via fog.

5.2.4 FEVM Structure

The proposed FEVM model comprises three main components for event validation: 1) ***Vehicle Confidence Level Module***, which permits each vehicle peer to form and propagate their self-confidence value regarding an event's occurrence, either through detection of the actual event or by forming event confidence through the aggregation of event reports received by peers; 2) ***Fog Confidence Level Module***, which oversees event validation based on the available information from vehicles and IoT devices in its zone, and neighbouring fog nodes; and 3) ***Fog Action Module***, in which fog combines evidence using the Dempster-Shafer rule for combination with importance factors, establishes a decision (i.e., "Event," or "No Event"), and notifies vehicles in its zone and neighbouring fog nodes about the decision, see Figure 5.2.

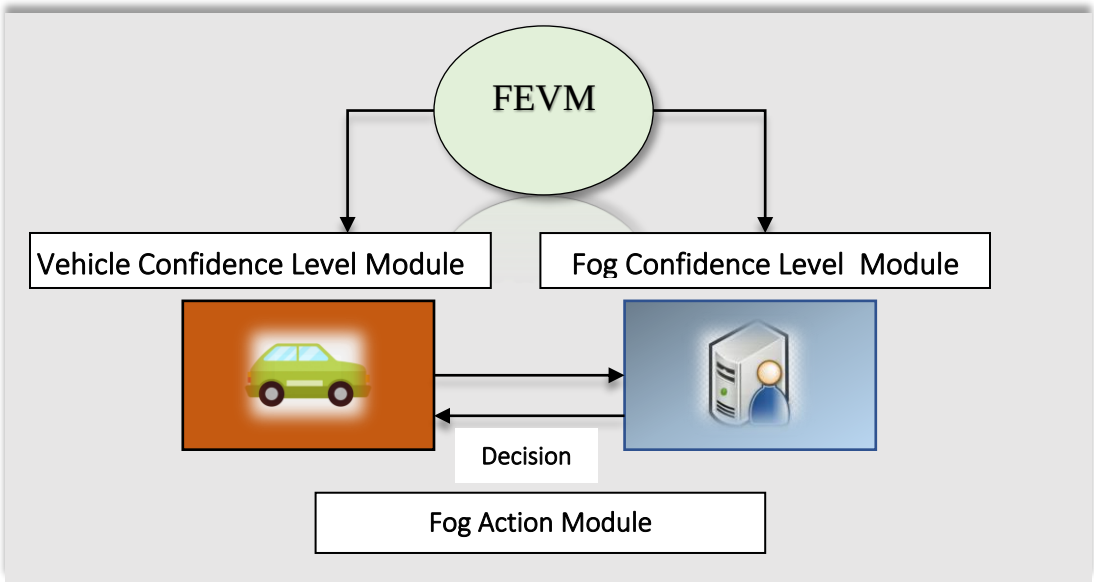


Figure 5.2: FEVM Components.

All vehicles in the VANET have several attributes considered in event validation. First, vehicles carry different levels of credibility based on their role type. In our model, we consider two role types: 1) High Role vehicles (HR) such as governmental vehicles or public transport vehicles: the reports from these vehicles are highly trusted; and 2) Low Role vehicles (LR) such as regular vehicles: the reports from these vehicles require greater support by evidence (e.g., high reputation value). Second, vehicles directly involved in event detection are distinguished from other vehicles involved in the event validation process. Report *generators* are vehicles directly involve in the event or are proximal to the event (e.g., within 50 m of the event) to permit direct detection of the event through vehicle sensors or biological senses of the driver. Report *propagators* are vehicles who are not *generators* but receive reports from other generators and/or propagators, perform an event confidence calculation on the report, and then transmit the updated report to fog and other vehicles within their transmission range. Because of the distinct nature of generators and propagators, they are factored in differently in confidence calculations. Third, each vehicle in our model has a predetermined reputation value based on its historical, verified accuracy in event confidence. This reputation value determines the vehicle's level of participation in event validation operations (i.e., the concept of TER in Chapter 4). Figure 5.3 shows the FEVM structure and how the vehicles, fog nodes cooperate to validate the existence of an event. Intra- and extra-layer communication occurs to validate an event's occurrence. Vehicles communicate with each other in the Edge Layer; fog servers and RSUs communicate within the Intermediate Layer; the Intermediate Layer is networked with the Edge Layer and the Main Layer to achieve cross-layer communication.

Note that we are here applying the same structure that we proposed in Chapter 4.

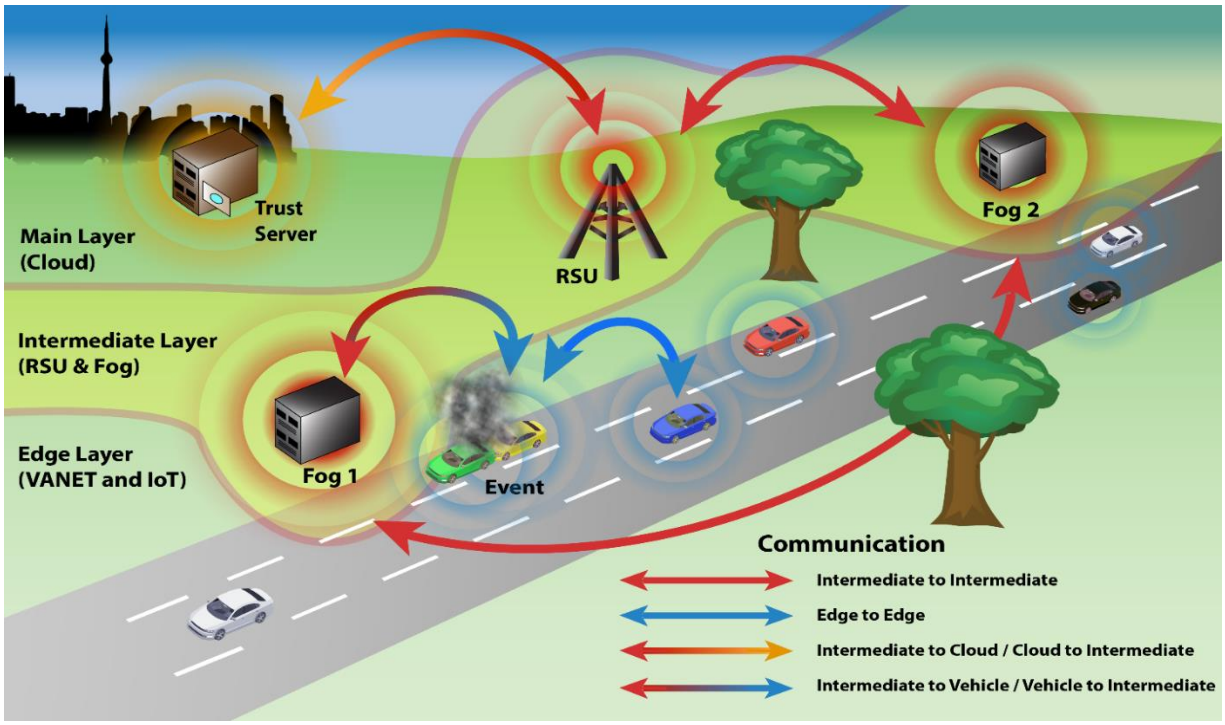


Figure 5.3: FEVM Structure and Event Validation involves communication within and across layers.

5.2.5 Vehicle Confidence Level Module

The Vehicle Confidence Level Module is responsible for computing confidence levels of reported events at the vehicle level. This module involves two processes for assessing confidence in reported events: 1) *Vehicle Class Integrity*, which is based on intrinsic attributes of the vehicle, and 2) *Aggregated Event Confidence Level*, which permits vehicles to assess their confidence in an event through the aggregation of event reports from generators and propagators. This module is designed to cover more cases that are not covered by RTEAM in Chapter 3. Unlike RETAM, which only covers the scenario where conflict reports exist, our design covers all possible scenarios where vehicles receive single reports or numerous reports, which may be unanimous or contradictory regarding the event's occurrence and may come from both generators and propagators.

A. Vehicle Class Integrity

The Vehicle Class Integrity, Γ, is assessed for all vehicles involved in the event validation process. Each vehicle's class integrity is based on intrinsic characteristics of the vehicle, driver, and timing of the report propagation:

1) The ability to detect the event by the driver's biological senses or the vehicle's sensors, is denoted ξ. $\xi \in [0,1]$, is a factor associated with the capabilities and reliability of the vehicle's onboard sensory system and the driver's historical reliability in manually registering the occurrence of events observed.

2) The vehicle's role (e.g., governmental vehicle, regular vehicle), is denoted T_r. This factor is defined by the role of the vehicle, and we use two categories, high reliability (HR) for governmental vehicles ($T_r = 0.9$) and low reliability (LR) for non-governmental vehicles ($T_r = 0.5$).

3) The time elapsed since the start of the event, relative to the event duration (for propagators only).

Because ξ is significant only for report generators, it is a factor in Vehicle Class Integrity for generators only. Thus, Vehicle Class Integrity is defined differently for generators and propagators. The Vehicle Class Integrity of ***generators***, Γ_g, is the product of ξ and T_r:

$$\Gamma_g = \xi \cdot T_r \quad \text{(Eq. 12)}$$

The Vehicle Class Integrity of ***propagators***, Γ_p, is defined as:

$$\Gamma_p = \begin{cases} T_r\left(1 - \frac{t_R - t_S}{E_d}\right), if\ (t_R - t_S < E_d) \\ 0, \qquad otherwise \end{cases} \quad \text{(Eq. 13)}$$

where t_R is the time that the event report is received from a propagator, t_s is the event timestamp (i.e., the starting time of the event), and E_d is the event duration.

Note that we assume that the event duration E_d, is automatically determined by the vehicle system (the generator's vehicle) according to the event type. For example, the event duration for a minor event (it is not a life-critical event) is 20 minutes, and the event duration for a major event such as an accident is 60 minutes.

✓ Illustrative Example

As an example, consider the case where a report generator is a government vehicle with a previously defined score for event detection of $\xi = 0.85$ based on the capabilities of the onboard sensors combined with the driver's history for reporting observed events. Since the vehicle is a government vehicle, T_r is set to 0.9 (for HR vehicle). The Vehicle Class Integrity for this vehicle is calculated as follows:

$$\Gamma_g = \xi \cdot T_r = (0.85)(0.9) = 0.765$$

Now, consider an event that starts at $t_s = 10$, and has one-hour duration. Consider a report propagator (A) and a report propagator (B) that are non-governmental vehicles propagating an event report 0.4h (24 minutes after the start of an event) and 0.9h (54 minutes after the start of an event), respectively. In this case, T_r is set to 0.5 (for LR vehicles), and the Vehicle Class Integrity for propagators A and B are calculated according to Eq. 14:

$$\Gamma_{\mathrm{p}}(\mathrm{A}) = T_r\left(1 - \frac{t_R - t_S}{E_d}\right) = 0.5\left(1 - \frac{24}{60}\right) = 0.3$$

$$\Gamma_{\mathrm{p}}(\mathrm{B}) = T_r\left(1 - \frac{t_R - t_S}{E_d}\right) = 0.5\left(1 - \frac{54}{60}\right) = 0.05$$

We can realize that the class integrity for propagator A is higher than the class integrity of propagator B due to the fact that propagator A is closer to the event location (it receives the report about the event earlier than propagator B). Now, assume that propagator C receives a report regarding the event after the event is expired; its class integrity will be 0.

B. Aggregated Event Confidence Level

A vehicle receiving report(s) from peers assesses confidence in the after receiving *n* reports from its neighbours. Confidence is assessed by aggregating the reports based on each of the propagators' confidence levels, C_{E_i}, reputation values, R_i, vehicle confidence Γ_i, and the number of hops the report made before receipt. The main purpose of relying on the peer confidence levels to validate an event and compute trust is to establish the confidence in the report sender on the accuracy of its report. Note that the peer of a vehicle includes any proximal vehicle within transmission range.

In this module, different methods to compute the peer confidence level are employed depending on the coherence of the reports. Receipt of reports may occur according to three possible scenarios: 1) Multiple Conflicting Reports (MCRs) are received- where *n* reports are received, and they are contradictory on the truth of the event's occurrence; 2) Multiple Unanimous Reports (MURs) are received - where *n* reports are received, and they are consistent on the truth of the event (i.e., all agree or all disagree the event occurred); or 3) a Single Report (SR) is received- the vehicle receives only one report on the truth of the event's occurrence.

Available trust data from peers is aggregated using a modification of the formula for aggregated trustworthiness from [88]. Here, we have modified the trustworthiness formula to replace peer trust values with the product of peer confidence and peer reputation. Further, we incorporate a penalty factor for hops, a penalty reduction for vehicle confidence, as well as a regularization factor that reduces confidence for smaller numbers of received reports on the event. We, therefore, name the calculated result *aggregated event confidence level*, C_E, to distinguish it from the calculation in [88]. A further modification to the calculation lies in the possibility of incorporating multiple report generators (the formula proposed in [88] assumes a single report generator exists). This accommodates events, such as car accidents or traffic congestion that could involve more than one report generator.

Vehicle confidence reported from generators is weighted more heavily (using a weighting factor, γ) in the aggregation computation since reports from generators are not reliant on the integrity of peers to establish confidence (i.e., it is not receiving a message from a peer or chain of peers; thus, there are fewer possibilities for corruption of the report; the only

possibility of corruption is for the generator to be corrupt). We assume that the report generators are trusted due to the fact that their reports are automatically produced by their onboard sensors.

- **Single Received Report from a Single Generator**

In this instance, a vehicle receives one report from a report generator. The event confidence level is calculated as follows:

$$C_E = \Gamma_{g_j} \alpha \quad \text{(Eq. 14)}$$

where Γ_{g_j} is the generator's vehicle class integrity, and $\alpha \in [0,1]$ is a regularization penalty based on having received only one report.

- **Single Received Report from a Single Propagator**

When a vehicle receives a single report from a propagator, the event confidence level is based on the senders' event confidence level and its reputation, while accounting for the additional hop in the information relay and the Vehicle Integrity Class of the propagator (i.e., the exponent of 1-Γ), the event confidence level is calculated as follows:

$$C_E = C'_{E_j} R_{e_j} \rho^{\left(1-\Gamma_{p_j}\right)} . \alpha \quad \text{(Eq. 15)}$$

where C'_{E_j}, R_{e_j}, and Γ_j are the propagator's event confidence level, experiential reputation score, and vehicle class integrity, respectively, and $\rho \in [0, 1]$ is a moderating factor, modulated by the exponent 1-Γ_j, which represents an additional hop minus the vehicle integrity class.

- **Multiple Received Reports from Multiple Generators**

In this instance, a vehicle receives multiple reports from generators only. The event confidence level is calculated as follows:

$$C_E = \frac{\sum_{i \in G} \Gamma g_i R_{ei}}{\sum_{i \in G} R_{ei}} \cdot \alpha^{1/n} \quad \text{(Eq. 16)}$$

where the summations are over all generators who have sent reports and *n* is the number of generators that have sent reports.

- **Multiple Received Reports from Multiple Propagators**

In this instance, a vehicle receives multiple reports from propagators only. The event confidence level is calculated as follows:

$$C_E = \frac{\sum_{j \in P} C'_{E_j} R_{e_j} \rho^{(1-\Gamma_j)}}{\sum_{j \in P} R_{e_j} \rho^{(1-\Gamma_j)}} \cdot \alpha^{1/n} \quad \text{(Eq. 17)}$$

where the summations are over all propagators who have sent reports.

- **Multiple Received Reports from Multiple different Senders**

When a vehicle receives multiple reports from both generators and propagators, the event confidence level is calculated as follows:

$$C_E = \frac{\gamma \sum_{i \in G} \Gamma_i R_{e_i} + \sum_{j \in P} C'_{E_j} R_{e_j} \rho^{(n_p-\Gamma_j)}}{\gamma \sum_{i \in G} R_{e_i} + \sum_{j \in P} R_{e_j} \rho^{(n_p-\Gamma_j)}} \cdot \alpha^{1/n} \quad \text{(Eq. 18)}$$

where γ is the weighting factor for report generators; Γ, C'_E, and R_e, are the vehicle class confidence, event confidence level, and reputation values, of reporters, respectively; $\alpha^{1/n}$ is the message count regularization factor; G is the set of the peers who are generators; P is the set of the peers who are propagators; $\mathrm{n_p}$ is the number of the received reports from propagators, ρ is the hop penalty factor; $\gamma > 1$, $R_e \in [0,1]$, $C_E \in (-1,1]$ and $\rho \in [0,1]$, and $\alpha \in [0,1]$. Note that there is a constraint $R_i \geq \tau$, where $\tau \in [0,1]$ is used as a reputation threshold customized by each peer or the network authorizers.

As shown in (Eq. 19), the peer confidence level is based on the aggregation of the vehicle confidence and peer reputation values of each sender. Moreover, the formula weighs the report generator(s) according to a weighting factor, gamma.

✓ Illustrative Example

To illustrate the computation Aggregate Event Confidence Level, consider a vehicle that has received multiple reports from a generator and propagators, as shown in Table 5.1.

Table 5.1: Five reports to be aggregated by receiving vehicle.

Vehicle ID	Reporter Type	C'_E	Γ	R
1	Propagator	0.838	0.40	0.9
2	Generator	NA	0.72	0.9
3	Propagator	0.629	0.45	0.7
4	Propagator	-0.310	0.40	0.7
5	Propagator	0.721	0.35	0.8

Further, assume parameter values γ, ρ, and α of 2, 0.8, and 0.9, respectively, are determined by the network authorizers for the VANET, and that receipt of five reports meets the threshold for aggregation. The receiving vehicle aggregates the five vehicle reports according to Eq. 18:

$$C_E = \frac{\gamma \sum_{i \in G} \Gamma_i R_{e_i} + \sum_{j \in P} C'_{E_j} R_{e_j} \rho^{(n_p - \Gamma_j)}}{\gamma \sum_{i \in G} R_{e_i} + \sum_{j \in P} R_{e_j} \rho^{(n_p - \Gamma_j)}} \cdot \alpha^{1/n}$$

$$= \frac{2 * (0.72 * 0.9) + [0.838 * 0.9 + 0.629 * 0.7 - 0.310 * 0.7 + 0.721 * 0.8] * 0.8^{(4-0.4-0.45-0.4-0.35)}}{2 * (0.9) + [0.9 + 0.7 + 0.7 + 0.8] * 0.8^{(4-0.4-0.45-0.4-0.35)}} * 0.9^{(1/5)}$$

$$= 0.6103$$

The receiving vehicle, therefore, calculates an event confidence level of 0.6103, and this value is to be transmitted in reports to all vehicles within range and to the nearest fog.

5.2.6 Fog Confidence Level Module

A. Fog Confidence Level

Fog combines evidence from all received reports after a threshold (i.e., threshold report number or duration since the beginning of the event), using Dempster' Rule of Combination with Importance Factors [86], which was proposed as an extension to the Dempster-Shafer Theory of Evidence. The Dempster-Shafer Theory of Evidence provides a theory for evidence and for probable reasoning and includes a rule for combining evidence to estimate probabilities of possible states. This theory was extended in [86] by incorporation of importance factors to lend different degrees of importance to different pieces of

evidence based on, for example, experience. The basic probability assigned to a state (or subset of states) *C* by combining evidence 1 and 2 is defined by:

$$m'^{(C,IF_1,IF_2)} = \frac{\sum_{A_i \cap B_i = C}\left[m_1(A_i)^{\frac{IF_1}{IF_2}} \cdot m_2(B_j)^{\frac{IF_2}{IF_1}}\right]}{\sum_{C \subseteq \theta, C \neq \varphi} \sum_{A_i \cap B_j = C}\left[m_1(A_i)^{\frac{IF_1}{IF_2}} \cdot m_2(B_j)^{\frac{IF_2}{IF_1}}\right]} \quad \text{(Eq. 19)}$$

where $m_1(A_i)$ is belief in the state or subset of states A by evidence 1, $m_2(B_j)$ is belief in the state or subset of states B by evidence 2, C is any non-empty subset of the state space, and IF_1 and IF_2 are the importance factors ascribed to evidence 1 and 2, respectively. For our purposes, the reference of discernment is constrained to mutually exclusive events E and E'. Under this circumstance, the rule for combining evidence can be re-written as follows:

$$m'(E_j, IF_1, IF_2) = \frac{m_1(E_j)^{\frac{IF_1}{IF_2}} \cdot m_2(E_j)^{\frac{IF_2}{IF_1}}}{m_1(E_j)^{\frac{IF_1}{IF_2}} \cdot m_2(E_j)^{\frac{IF_2}{IF_1}} + m_1(E'_j)^{\frac{IF_1}{IF_2}} \cdot m_2(E'_j)^{\frac{IF_2}{IF_1}}} \quad \text{(Eq. 20)}$$

We treat each vehicle's event confidence value as the belief in the occurrence of the event. First, *C* is converted to an interval [0, 1],(i.e., normalization N) by:

$$NC'_{E_i} = \frac{C_{E_i} + 1}{2}$$

For each vehicle report. Fog then ascribes a probability to each vehicle report to for event state E

$$m_i(E) = NC'_{E_i}$$

We base the importance factor for each evidence on a vehicle's reputation relative to the *n* vehicles reporting:

$$IF_i = \frac{R_{ei}}{\sum_{j=1}^{n} R_{ej}} \quad \text{(Eq. 21)}$$

Thus, if vehicle *i*, with reputation value R_i, provides a report to fog with $C_{E_i} > 0$, then vehicle *i* provides evidence of event E with basic probability for E equal to C_{E_i} and this evidence is associated with an importance factor IF_i .

More than two vehicle reports can be combined using the Dempster-Shafer rule for combination as follows:

$$m_1 \oplus m_2 \oplus m_3 \oplus \ldots = (((m_1 \oplus m_2) \oplus m_3) \oplus \ldots)$$

where $\oplus$ is the Dempster's rule for combination with importance factors and the importance factor assigned to the result of each combined pair is (IF1 + IF2)/2. For example, if we combine evidence from vehicles V1 and V2 with beliefs m_1 and m_2 about the event state then

$$V_1\langle m_1, IF_1\rangle \oplus V_2\langle m_2, IF_2\rangle \rightarrow V_{combined}\langle m_1 \oplus m_2, (IF_1 + IF_2)/2\rangle$$

The resultant evidence, $V_{combined}$, can then be treated as a single piece of evidence and combined with another using Dempster's rule for combination. This process can be iterated to combine any number of vehicle reports.

✓ Illustrative Example

Suppose a fog server receives reports from three vehicles in its zone, as summarized in Table 5.2.

Table 5.2: Evidence to be combined from three vehicles' reports.

Vehicle ID	C_{E_i}	R
1	0.9	0.8
2	0.7	0.9
3	0.5	0.7

The basic probability assignment for the event based on each report i is calculated as follows:

$$m_i(E) = \frac{C_{E_i}+1}{2}$$

$m_1(E) = 0.95$; $m_2(E) = 0.85$; and $m_3(E) = 0.75$.

Relative importance factors are then assigned to each report using Eq. 21. IF_1 is calculated for the first report:

$$IF_1 = \frac{R_{e_1}}{\sum_{j=1}^{3} R_{e_1}} = \frac{0.8}{0.8 + 0.9 + 0.7} = 0.3333$$

Similarly, IF_2 and IF_3 are calculated, and we get the following:

$$IF_2 = 0.3750, \text{ and } IF_3 = 0.2917.$$

We now have values for each vehicle report, as shown in Table 5.3.

Table 5.3: Basic probability assignments and calculated importance factors for three vehicle reports.

Vehicle ID	m_E	IF
1	0.95	0.3333
2	0.85	0.3750
3	0.75	0.2917

The Extended Dempster-Shafer rule for combination is used to combine reports, two at a time. Beginning with the reports for vehicles 1 and 2, the Dempster-Shafer rule for combination (Eq. 20) is used as follows:

$$m'(E, IF_1, IF_2) = m'(E, 0.3333, 0.3750) = \frac{(0.95)^{\frac{0.3333}{0.3750}} \cdot (0.85)^{\frac{0.3750}{0.3333}}}{(0.95)^{\frac{0.3333}{0.3750}} \cdot (0.85)^{\frac{0.3750}{0.3333}} + (0.05)^{\frac{0.3333}{0.3750}} \cdot (0.15)^{\frac{0.3750}{0.3333}}}$$

$$= 0.9897$$

The importance factor for vehicles 1 and 2 combined is also summarized:

$$V_{1\&2\ combined}\langle m'(E, 0.3333, 0.3750), (IF_1 + IF_2)/2\rangle \rightarrow V_{1\&2\ combined}\langle 0.9897, 0.35415\rangle$$

Evidence from vehicle report 3 is then combined with $V_{1\&2\ combined}$ to form a probability assignment the combined evidence from all three reports:

$$m'(E, 0.2917, 0.35415) = \frac{(0.95)^{\frac{0.2917}{0.35415}} \cdot (0.85)^{\frac{0.35415}{0.2917}}}{(0.95)^{\frac{0.2917}{0.35415}} \cdot (0.85)^{\frac{0.35415}{0.2917}} + (0.05)^{\frac{0.2917}{0.35415}} \cdot (0.15)^{\frac{0.35415}{0.2917}}} = 0.9941$$

Consequently, the probability assigned by fog to event E being true based on the combined evidence of the three vehicles' reports is 0.9941. This value is passed to the Fog Action Module and in the meantime, Fog continues to receive additional evidence until a revision threshold is met, at which point a new probability is assigned to event E, by combining all evidence.

B. Fog Action Module

The Fog Action Module decides on the event with the largest probability assignment. If *m*(E) is greater than 0.5 + ϵ, then the Action Modules decides E, otherwise E'. An error threshold, ϵ, is nominally set to 0 but can be increased to reduce false positives or reduced to reduce false negatives for situations where we are more averse to one type of error over the other. Once the decision is made, fog promulgates the decision to all vehicles in its zone and to all neighbouring fog nodes, along with appropriate action for the vehicles.

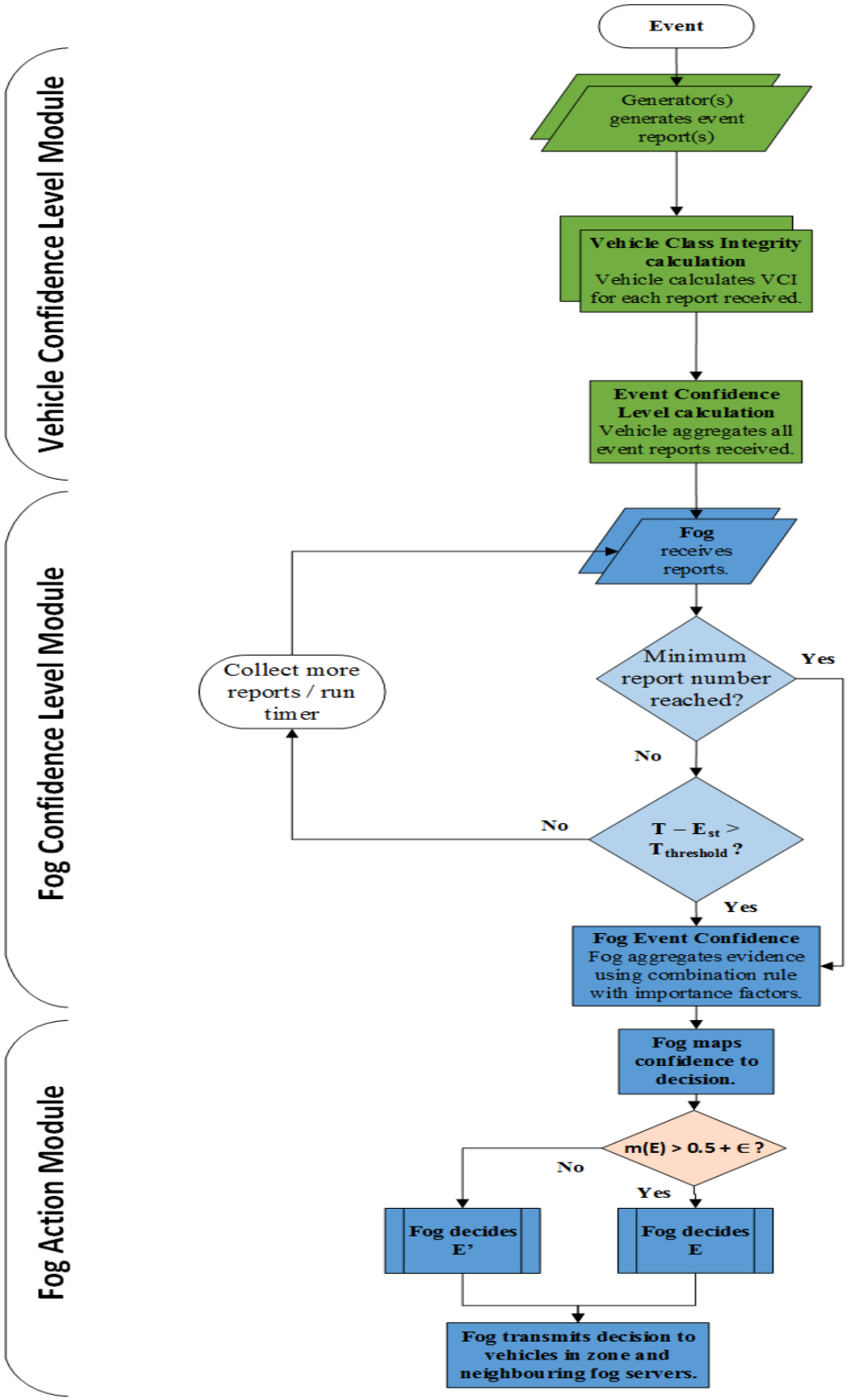


Figure 5.4: FEVM flowchart.

5.2.7 How FEVM works?

This section describes the FEVM process, beginning with event occurrence and culminating in action decided by fog. The steps described below are captured in the flow chart of Figure 5.4.

Event Reporting

1. If an event occurs (or not), it may be detected (fabricated) by the vehicles involved in the event, by passerby vehicles, or by IoT devices along the transportation route. Once detected, a report(s) is generated by the detecting party (one or more of the abovementioned entities).
2. The report is transmitted to nearby vehicles and to the fog server overseeing the zone.

Vehicle Confidence Level Module

3. Vehicles receiving report(s) immediately calculate the sender's Vehicle Class Integratity based on previously determined vehicle role and event detection abilities (in the case of generators) along with temporal values for the event (event start, event duration, current time).
4. Vehicles calculate the Event Confidence Level, as described in Section 5.2.5, confidence in a generator's report is based on the Vehicle Class Integrity of the generator, and the generator's reputation score.
5. The vehicle sends a report, including the Event Confidence Level, to the local fog server.

Fog Confidence Level Module

6. A threshold (based on time elapsed since start of the event or receipt of a minimum number of vehicle reports) or revision threshold (based on additional time elapsed or receipt of additional reports) is met for calculating (or revising) the probability assignment, *m(E)*.
7. The Fog Confidence Module combines evidence using the Dempster's rule of combination Importance Factors (Section 5.2.6) to calculate the probability assignment, *m(E)*. This is the overall confidence in the occurrence of the event.
8. The probability assignment *m(E)* is passed to the Fog Action Module.

Fog Action Module

9. Fog makes a decision based on the probability assignment, for event E. If $m(E) > 0.5 + \epsilon$, then fog decides E, otherwise E'.
10. Fog selects the appropriate action based on the decision (Fog Action Module, Section 5.2.6).

11. Fog transmits the decision and action of the Fog Action Module back to all vehicles in the Edge Level of the VANET and to neighbouring fog servers. This transmission also includes what action(s) to take.

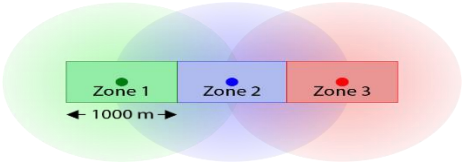
5.3 Performance Evaluation

In this section, we use ns-3 to simulate and evaluate our proposed model. For the simulation, a 3000 m stretch of 3-lane highway is divided into three (fog) zones. Fog servers are positioned in the center of their respective zones, and their communication ranges extend beyond their zone boundaries such that the servers can communicate with one another (i.e., the fog transmission ranges are at least 1000 m).

Once the simulation is started, two vehicles are fed into the transportation route at 1 sec and additional vehicles are fed into the transportation route every 1-1.5 sec thereafter. Vehicles enter the transportation route in Zone 3 and proceed through Zone 2 and then Zone 1. Each of the two vehicles involved in the accident transmits event reports to Fog 1 and continuously to any vehicles travelling within their transmission ranges (100 m is the assumed range of V2V communications [81]). Reports from event report generators are set to be trusted. We assume that all reports are valid and all senders' trust values exceed the trust threshold. In each zone, the fog server receives reports from the vehicles in its zone and from neighbouring fog servers. In our simulation, we assume that only fog in zone 3 only has to make a decision since it's the fog server that welcomes the newcomer vehicles, and it can warn them about the event. Thus, Fog in zone 1 sends received reports to fog in zone 2, which forwards the report to fog in zone 3, and fog in zone 3 decides the event state.

For our simulations, we vary the percentage of malicious nodes, which falsely purport the accident did not occur. The main simulation parameters are shown in Table 5.4.

Table 5.4: Simulation parameters.

Parameter	Value
Simulation transportation route	3-lane, 1-way highway
Wireless communication interface	IEEE 802.11P
Transportation route size and layout	3000 m The area is divided into 3 zones Each zone is 1000 m and is served by fog servers with overlapping transmission ranges.  The dots represent the 3 fog servers. The fog transmission range = **1000 m** (represented by the larger circles).
Simulation duration and timing	**Overall Simulation duration**: 200 sec **Accident Start t_s**: 80 sec **Event Duration E_d**: 100 sec Accident zone: **Zone 1**
Average vehicle speed	**80 km/h**
Total number of vehicles (N_v)	**100**, distributed as follows: ▪ **10** (10%) are GV ▪ **78** (78%) are NGV ▪ **2** (2%) are NGV which crash
Vehicle transmission range	**100 m**
Malicious vehicle percentage (they are NGV)	**5, 10, 20, 30, 40, 50%**
Vehicle role (T_r)	➢ **0.9** for government vehicles GV ➢ **0.5** for NGV
Reputation (R_e) of Governmental Vehicles (GV)	**0.7**
Reputation (R_e) of other Non-governmental Vehicles (NGV)- (Honest)	**0.7**
Reputation (R_e) of Non-governmental Vehicles (NGV) (Malicious)	**0.6**
Calculation constants for the simulation	**$\alpha = 0.9$** **$\rho = 0.9$** **$\gamma = 2.0$**
Malicious vehicle model	lying about the existence of an event -always deny the event's existence
Fog decision	**E** – EVENT **E'** – NO EVENT

5.3.1 Fog Decision Threshold

The fog decision threshold is based on having reached a minimum number of reports or a maximum allowed time (whichever occurs first). We simulated the FEVM using four different fog decision thresholds, specified as follows:

- ***Scenario 1***: The decision threshold for fog is set to receipt of 10 reports from 10 different vehicles (10% of vehicles) or 100 sec since the event start, whichever occurs first. (20 sec after the event starting)
- ***Scenario 2***: The decision threshold for fog is set to receipt of 20 reports from 20 different vehicles (20% of vehicles) or 110 sec since the event start, whichever occurs first. (30 sec after the event starting)
- ***Scenario 3***: The decision threshold for fog is set to receipt of 40 reports from 40 different vehicles (40% of vehicles) 120 sec since the event start, whichever occurs first. (40 sec after the event starting)
- ***Scenario 4***: The decision threshold for fog is set to receipt of 60 reports from 60 different vehicles (60% of vehicles) or 130 sec since the event start, whichever occurs first. (50 sec after the event starting)

TPR is used to evaluate the performance of FEVM with the four above mentioned decision thresholds, where TPR is defined as follows:

➢ **TRUE POSITIVE RATE (TPR):** represents the probability of correctly detecting the event state (i.e., fog decides E).

The TPR is for each scenario with various malicious vehicle prevalence is shown in Figure 5.5.

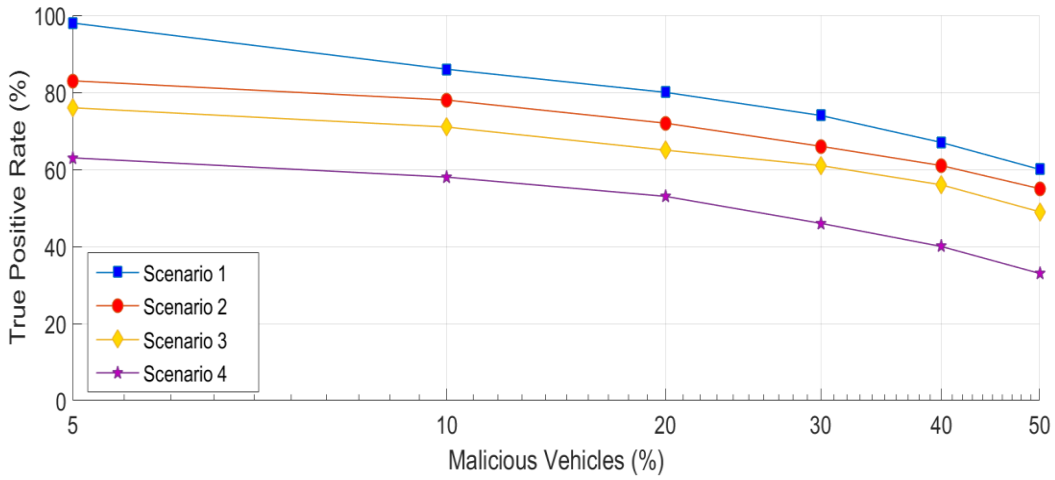


Figure 5.5: True Positive Rate for the four FEVM scenarios.

Predictably, increasing the malicious vehicle prevalence reduced the TPR for all scenarios since this corresponds to a greater number of false reports in the evidence collection. Also, the TPR is higher for smaller decision thresholds (i.e., lower threshold resulting in earlier decision) because in our simulations, the initial report generators are honest vehicles that observed the event

directly, and the malicious vehicles enter the transportation route over time (i.e., the initial reports are correct and there are a low number of malicious vehicles early in the simulation). Thus, progressing from Scenario 1 through Scenario 4 reveals a decremental effect on TPR associated with the incremental decision threshold.

5.3.2 Comparison with RTEAM

To show the effectiveness of FEVM, we compare it with RTEAM in (the model from Chapter 3) accordance with the TPR metric:

Each RTEAM version is compared to the four FEVM scenarios in Figures 5.6, 5.7, and 5.8.

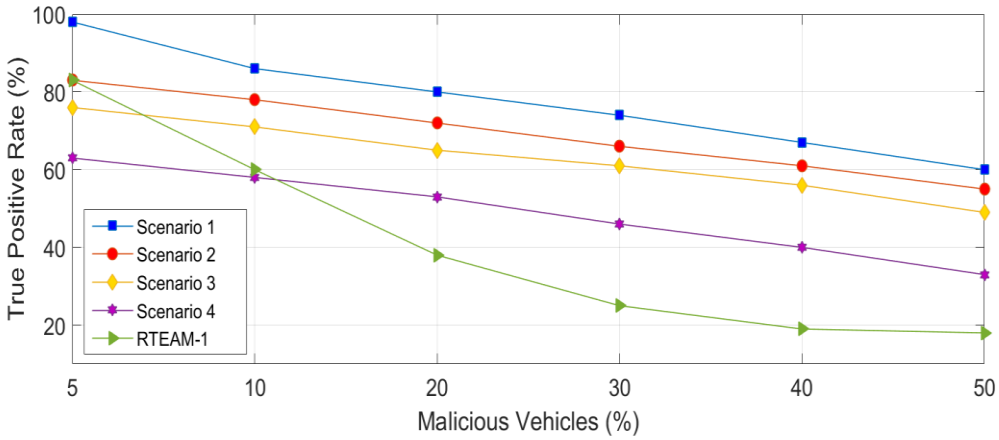


Figure 5.6: True Positive Rate for four FEVM scenarios VS. RTEAM-1.

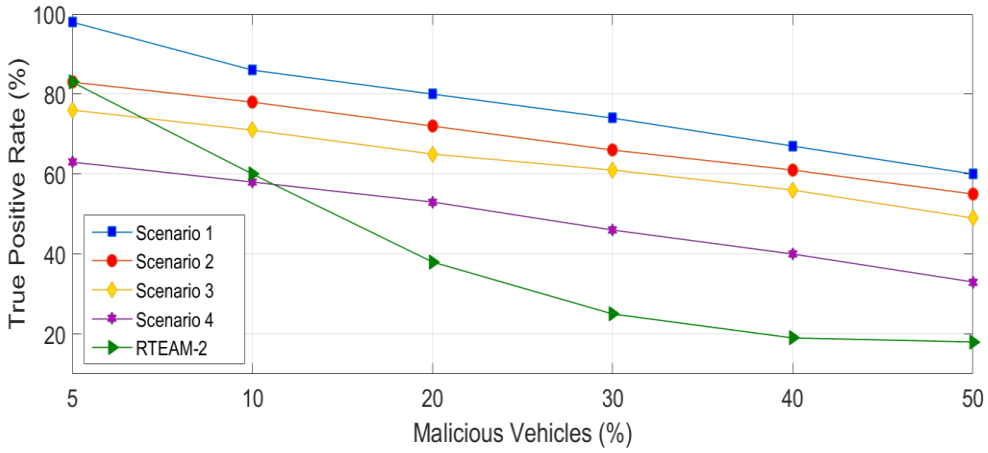


Figure 5.7: True Positive Rate for four FEVM scenarios VS. RTEAM-2.

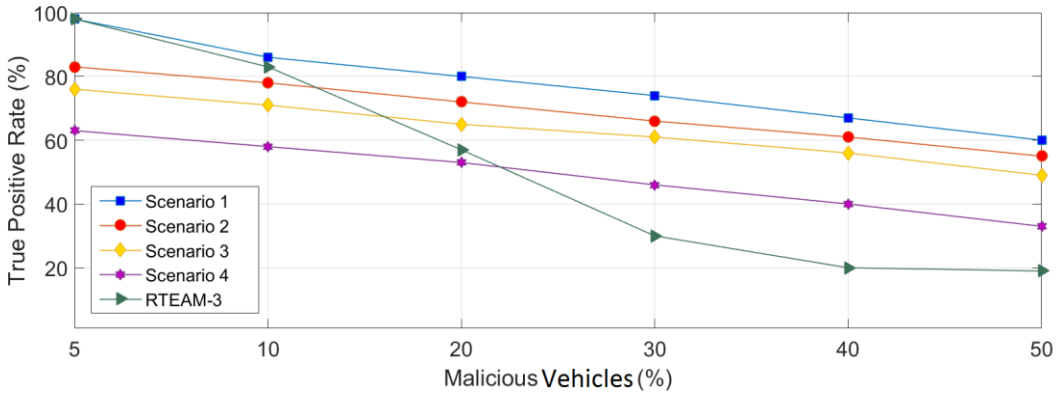


Figure 5.8: True Positive Rate for four FEVM scenarios VS. RTEAM-3.

In Figure 5.6, at 5% malicious vehicle prevalence, RTEAM performs at least as well as all FEVM scenarios except FEVM Scenario 1. The RTEAM TPRs also drop off with the malicious vehicle prevalence; however, they drop off more steeply than the FEVM TPRs and, at 30% malicious vehicle prevalence, RTEAM is underperforming all FEVM scenarios with respect to TPR. It must be noted, however, that RTEAM sacrifices TPR to reduce risk in some circumstances. In these plots, we can see that RTEAM-2 and RTEAM-3 have similar performance. This is because these scenarios have similar parameters, and comparable error intensity curves were specified for RTEAM-2 and RTEAM-3 (as shown in Figure 3.3 and Figure 3.4, respectively).

5.3.3 Comparison with other Trust models

To show the effectiveness of FEVM, we compare FEVM with: 1) Our model RTEAM in Chapter 3, 2) Simple Trust Model (STM), where the earliest report is followed, 3) Hop-based Trust Model (HTM) [19], and 4) Multi-faceted Trust Model (MTM) [43]. For simplification purposes, we choose the best RTEAM version. We defined the following metrics to evaluate the efficiency of FEVM:

- **UNDEFINED CASES (UND)**: represents the number of cases that the model 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 (i.e., node decides E).

➢ **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 5.9 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 50%. 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 that 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 the decision in the way that the vehicle cannot identify the event state (i.e., the weight of the vehicles agree with the occurrence's of the event mines the weight of the vehicles disagree with the occurrence's of an event is equal zero). With increasing the malicious in the network (i.e., most of the participants deny the occurrence's of the event), the UND cases are decreased (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 and FEVM consider different aspects of the senders and rely on serval trust metrics before deciding the action regarding an event state. Regarding MTM and HTM, it can be seen that RTEAM and FEVM 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.

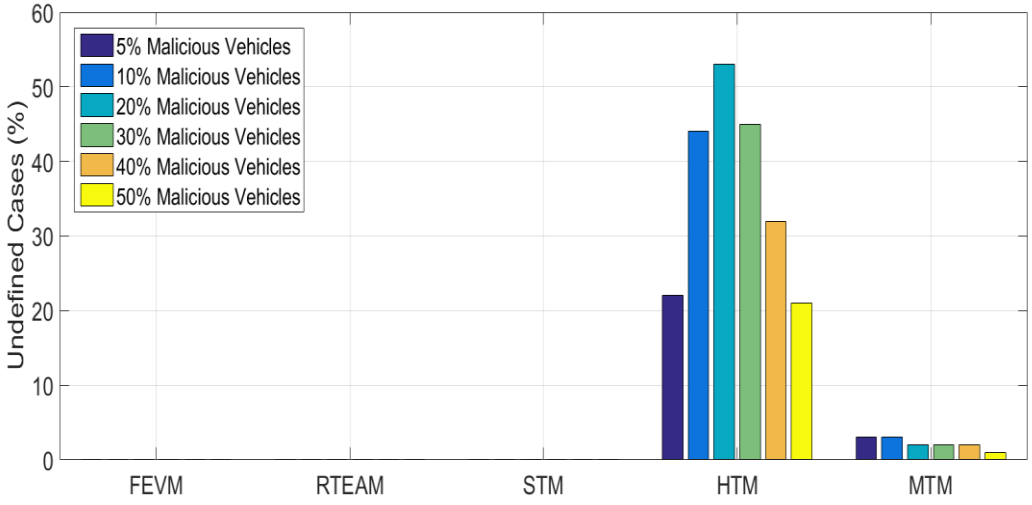


Figure 5.9: Undefined Cases (UND) for FEVM VS. STM, MTM, and HTM.

➢ **TRUE POSITIVE RATE (TPR)**

Here, the TPR performance of the best-case scenario (Scenario 1 indicates lowest decision threshold) and worst-case scenario (Scenario 4 indicates the highest decision threshold) of the FEVM simulations are compared to the STM, MTM, and HTM models. As shown in Figure 5.10, the FEVM achieves higher TPRs in Scenario 1 compared to the other models at all malicious vehicle prevalence's. For Scenario 4 (see Figure 5.11), the FEVM outperforms the other models for all malicious vehicle prevalence's except for 5%, at which it performs better than only STM. With 5% malicious vehicles, HTM and MTM show better TPRs because, in FEVM, the fog has to wait a certain time and involve more reports compared to HTM and MTM, which increases the probability of receiving fake reports. FEVM has a higher TPR (with 5% malicious vehicles) compared to STM because STM makes its decision based only on a single report (the earliest report), which may negatively affect its performance. However, FEVM outperforms the other models in terms of TPR by increasing the percentage of malicious vehicles in the network (more than 5%).

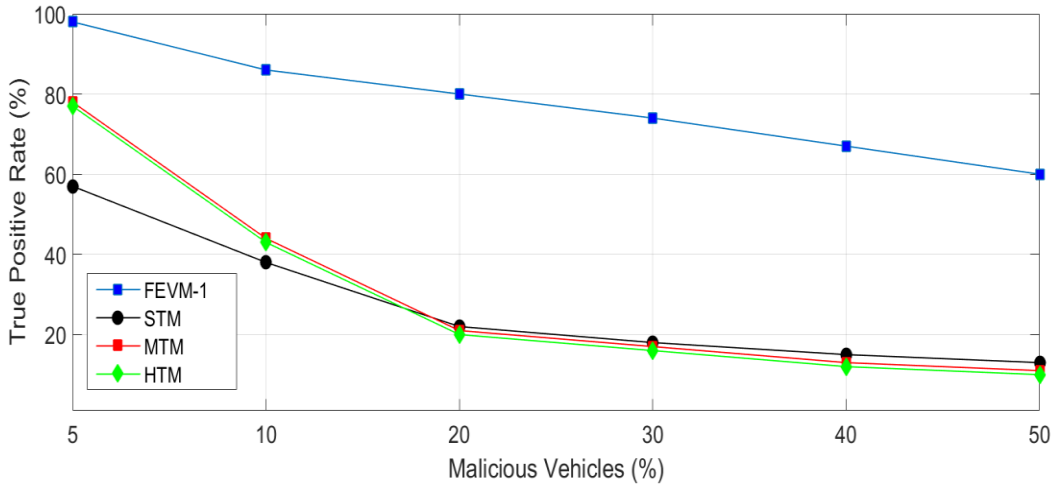


Figure 5.10: True Positive Rate for FEVM (the best scenario) VS. STM, MTM, and HTM.

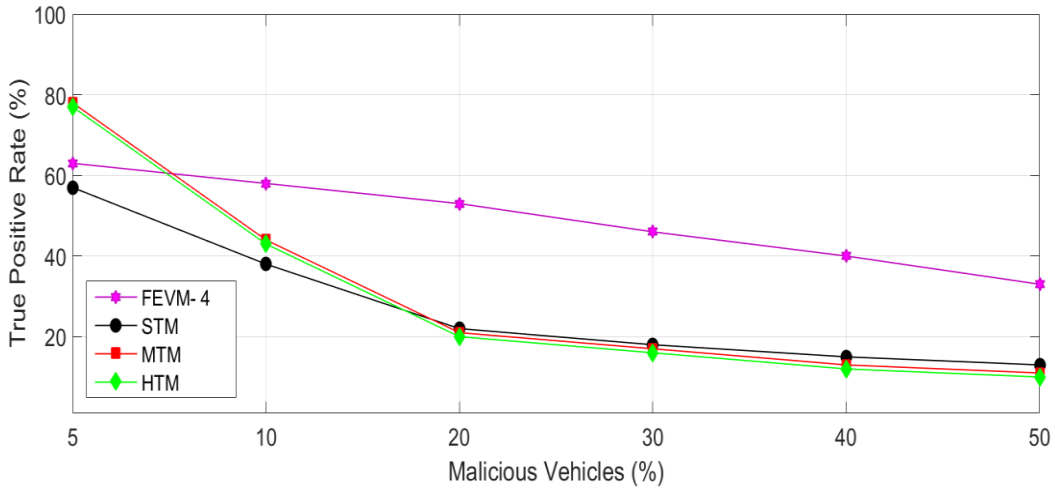


Figure 5.11: True Positive Rate for FEVM (the worst scenario) VS. STM, MTM, and HTM.

5.4 Discussion

- ***Fog Confidence Module***

Fog does not require reports strictly from vehicles to implement the confidence and action modules. Fog may also receive reports from IoT devices in its zone, such as cameras, or from a neighbouring fog server. The fog server overseeing the reporting IoT device could propagate the message to neighbouring fog servers, which in turn warn vehicles along the transportation route until the event has expired. As with vehicles, reports from IoT devices could be made in error. The vehicle's sensors are fallible and could fail to detect an event (i.e., detects "No event") or become hijacked. Confidence in reports from IoT devices can be based on attributes such as technology type, sensitivity/specificity of the device, history of successful detection, redundancy of detectors, etc. Reports from IoT devices can be aggregated (along with reports from vehicles) in a manner similar to that of Vehicle Confidence Module in Section 5.2.2. Although not detailed in this work, these scenarios would require a minimum number of reports or passage of time to meet a threshold at which time a decision is forced, similar to the process described for vehicle reports in Section 5.2.3, and should be considered in future work.

In our simulations, the model is evaluated based on decisions made by Fog 3, overseeing the zone in which the event occurred. In this scenario, Fog 3 makes a decision and notifies all vehicles in its zone of the accident. Future work should consider how fog servers in other zones (Fog 1 and Fog 2 in our simulation) behave. These fog servers could make decisions based on the information available to them (the same behaviour as Fog 3) or they could refrain from decision-making and only transmit warning messages to vehicles in their zones of a possible event ahead on the transportation route.

A time threshold (a time expiration since the event start) at which fog is forced to aggregate and decide on the event occurrence is incorporated in our work. An optional strategy is to make this threshold dynamic, based on factors such as the event type (major, minor), event location, traffic density, weather conditions, etc. In cases where fog is delayed or fails to send a decision within a maximum permitted time, it would be advantageous for vehicles to make decisions and take evasive actions based on their own calculated event confidence levels. When a vehicle transmits an event report to fog, a timer is started. If fog fails to respond with a decision within a set time, the vehicle may assume the decision-making process. Such a vehicle module would therefore guarantee that a decision will be made within a reasonable to time avoid collisions and increase road safety in future work.

We employed the Dempster-Shafer rule for combination using importance factors, as developed in [86] to combine evidence in the Fog Confidence Module. As pointed out in [89] one issue that may arise using this rule occurs when combining conflicting evidence that happens to intersect only for a state(s) of very low probability. In such cases, it is possible that the rule for combination will assign a high probability to the unlikely state because it is the only state assigned non-zero probability in both pieces of evidence. This issue has been addressed in many works, including in research specifically dedicated to trust calculations VANETs [42]. Our work avoids this issue because we consider only dichotomous frames of discernment (i.e., the event space has only two, complementary outcomes) in which both outcomes have non-zero probability due to the constraints of our model (i.e., all reports from generators or propagators have non-zero probability assigned to both outcomes). Therefore, the described issue is avoided because the intersection of all evidence includes both E and E'. Future extensions to our model, however, should expand beyond the binary frame of discernment to permit more than two states. For example, if a generator reports an event, but passersby generate reports purporting a different type of event, then the frame of discernment may expand to three or more possible states (e.g., 1. Major accident; 2. Minor accident; 3. No event).

- ***HR Vehicles***

We have assigned a weight, γ, to report generators, emphasizing their importance in event confidence. A possible extension to this concept is to fully trust any HR report generator. We can trust HR reports only if these vehicles have a direct connection with IoT devices and hear from them. Under this scheme, if fog receives a report from an HR generator, no computation is necessary. Fog simply assumes E and transmits the decision and actions throughout the VANET. With respect to HR vehicles acting as propagators, their higher integrity (Vehicle Class Integrity) is less significant because their propagated reports inherit the uncertainty already inherent in the chain of propagation preceding them, regardless of their integrity. Therefore, the Vehicle Class Integrity is used only to modulate the moderating factor, ρ, for propagators.

5.5 Conclusion

The FEVM continues the theme of centering fog in the event validation process. In this model, we introduce some flexibility in combining vehicle reports from a spectrum of sources by use of importance factors. Since information received from different sources will carry different levels of trust, confidence, reliability, etc., the FEVM draws on an extension to the Dempster-Shafer Theory of Evidence (DSE) which permits fog to combine evidence from these sources while ascribing different importance factors to each. In this model, vehicles assessment confidence levels in the event using more conventional methods of aggregation. Reports from vehicles received by fog are then combined using the DSE with importance factors. A key feature of the FEVM is imposition of a singular decision (and associated actions) by fog to all vehicles in its zone, which promotes coherent behaviour.

The FEVM was simulated with different thresholds for promptness of decision and with different malicious vehicle prevalence. Based on the TPR metric, the simulations showed that when report generators were honest (i.e., the malicious vehicles could only be propagators) performance was best when thresholds were set to shorter duration / fewer reports. This occurred because malicious vehicles had less time to corrupt information in the VANET. Intuitively, higher malicious vehicle prevalence reduced the TPR in all scenarios. When compared to RTEAM, FEVM was comparable (in terms of TPR) at 5% malicious vehicle prevalence but generally outperformed RTEAM at prevalence's of 20% greater. It is acknowledged that RTEAM sometimes trades off TPR in favour of reduced risk, which gives FEVM an advantage with respect to TPR.

Chapter 6:
Conclusion & Future Work

6.1 Conclusion

With over a billion vehicles in use worldwide, traffic fatalities have recently numbered greater than a million per year. Furthermore, these numbers are expected to approximately double over the next 15 years [2]. VANETs constitute a promising technological development for mitigating the risks of vehicular travel, as well as for improving transportation efficiency and occupant comfort. These enhancements are made possible in VANETs by trustworthy, secure, and private radio communication between vehicles and infrastructure aimed at conveying safety, traffic, and other useful information.

Transmission of information in any network is only useful insofar as it is trusted. The information age has demonstrated the prevalence of malicious behaviour on networks and its potential, severe repercussions. Thus, a cardinal design goal of VANETs is a robust trust evaluation mechanism for the information communicated throughout the network. Dedicated to this purpose, numerous models have been proposed to evaluate trust or reputation among VANET users and the information they transmit. These models have their relative merits. In this work, we aim to address various challenges faced by VANETs, including communication overhead, the computational workload of nodes, overly general trust metrics, and lack of risk assessment. With this objective, we propose three models with an underlying theme of promoting fog as the key pillar of trust management and event validation in the VANET. We also introduce novel concepts of task-specific trust metrics and risk-informed decision-making.

6.1.1 Risk-based Trust Evaluation Advanced Model (RTEAM)

Conventionally, decision-making in VANETs is conducted by assessing for which state there exists the greatest trust or confidence and then deciding on that state as the true state, irrespective of the consequences of a wrong decision. These consequences could be innocuous or catastrophic, but since they are not considered in conventional models, they do not affect decision-making process. Holding safety as a prime objective, decisions must integrate consequence with trust to evaluate risk, the product of probability and consequence. Our first model, RTEAM, overcomes serious limitations of conventional VANETs by using risk minimization, rather than trust, to drive the decision-making process. The backbone of risk-based decision-making is the quantification of risk as the product of trust and consequence. Specifically, the decision-making module computes the product of the probability of making a wrong decision and the consequence of the wrong decision to arrive at a risk score. The

quantified consequence component of the risk-based model is established on metrics of closeness to the purported event (i.e., time available to take the appropriate action), the closeness and the density of neighbours that will be affected by the vehicle's decision. We assessed our proposed model under extensive experimentation and demonstrated its effectiveness by comparison with a pure trust model in terms of true positive rate, risk level, and the number of cases that cannot be recognized by the model.

Simulated results demonstrate the risk-based model outperforms a pure trust-based model in terms of incurred risk levels. This improvement in performance results because the risk-based trust model always seeks the lowest-risk action. Sometimes, this involves deciding on a state for which there is lower confidence but nevertheless minimizes risk. On the other hand, the trust-based model decides upon actions based on the highest trust values of reports and does not heed risk.

RTEAM is a lightweight model that reduces processing time, saves resources, and conserves energy by adopting a two-phase filtering scheme by screening out invalid reports (1st phase) and reports that do not meet authentication and trust criteria (2nd phase). Unlike simple voting and majority-based methods (which cannot execute decisions when believers and deniers are equal and the vehicles with the highest similar experience/role-based trust have different opinions), RTEAM can estimate the event state at any time. The ability to reduce potential consequences is an important contribution of this model because safety is a key objective of VANETs. Traditional VANET models based on trust or confidence assessment, though ostensibly seeking to minimize adverse consequences, may inadvertently expose network members to greater risk by making decisions based solely on trust and irrespective of the potential consequences of the decision. The RTEAM model, therefore, opens the door to future exploration of risk-based assessment and decision-making in VANETs.

The (dynamic) nature of the consequences of wrong decisions, externalized in the error intensity curves, is undeveloped in this work and remains conceptual. However, the success of RTEAM depends on a proper definition of the error intensity curves; improperly characterized curves could lead to undesirable results.

The proposed model uses trust consensus as a proxy for likelihood (α and β) in the risk calculation. More thorough estimates for likelihood may be derived by incorporating other forms of confidence and information from other sources, such as IoT devices, other fog servers, etc., and should be considered in future research. Our research may be built on by implementing more comprehensive simulations of diverse scenarios and through further comparison of performance against existing models. RTEAM mainly has built based on MTM, which affects its performance.

6.1.2 Fog-based Reputation Evaluation Model (FREM)

Our second model, Fog-based Reputation Evaluation Model (FREM), capitalizes on fog's position in the intermediate layer to reduce communication overhead and workload born by vehicles nodes in the VANET. In FREM, fog nodes are considered the main points that: 1) provide capacity to the VANET (i.e., storage and computation) that reduces the deployment of other expensive infrastructure such as RSUs and reduces the computational workload on the nodes (i.e., no need for real-time experience). Fog nodes are used as DTC recordkeepers in the short and long-term; 2) use DTCs to reduce message complexity of the trust establishment; 3) identify preferred paths that vehicles travel daily or weekly are to learnmore about the most visited vehicles that have better reliability and knowledge of the transportation route.

A key development brought forth in our FREM model is the strategy of Task-based Experience Reputation (TER). Under TER, trust values are assessed specific to the task type. Thus, each node has multiple trust values, each representing the experiential success of a specific task type (i.e., a task-based trust profile is developed for each node). This has a distinct advantage over conventional models that represent trust with a single value, regardless of the task. Using conventional models, a vehicle that has performed poorly for a given task type may nevertheless have a high trust score based on successful performance of unrelated tasks. In this situation, the VANET is blind to the distinctions unveiled by task-based trust and can only assess based on the single trust score. Such models, therefore, have "all in the same boat" when making trust-based decisions. In our model, fog develops a detailed understanding of each node's strengths and weaknesses and can therefore select the node(s) best suited for the task at hand. We refer to this simple modification as *smart employment* in the VANET. This method has significant benefits when considering safety-related tasks versus non-safety-related tasks. In matters of safety, it is imperative to select the vehicle most trusted for the safety-related task – not necessarily the one with the highest overall trust. Without TER, malicious nodes can bolster their trust scores through successful completion of benign tasks and then sabotage a safety-significant task assigned to them. A further contribution of this model is the local storage of trust in the intermediate layer. Whereas existing models involve computation of trust vehicles in the edge layer and deletion of the value from the vehicle's memory immediately after taking action, this model permits access by fog to historical trust records, which constitute a rich source of information for evaluation that is more accurate and reliable.

Simulations of this model showed a reduction in message transmission overhead (e.g., the FREM shares the workload with RSU to reduce the vehicle workload to 50% or less), a reduction in the magnitude of the workload, and effectiveness of task-based reputation values.

Under TER, as the distinction of the task types on which trust values are based becomes more precise, the amount of trust data per task type necessarily becomes smaller (i.e., the amount of data for a specific task is smaller when the number of evaluated task types is larger). Thus, a balance between task resolution and statistical power must be struck for the purposes of

evaluating task-based trust. When information is scarce or absent, there may be no trust established for a specific task type. A further limitation of this model is that it does not revise trust values in real-time, but incurs a lag, since trust value updates to the Trust Server may take up to 24h. Efficient means of updating the Trust Server quickly should be considered in future work.

6.1.3 Fog-based Event Validation Model (FEVM)

The proposed FEVM model extends the fog-centric theme by appointing decision-making authority to fog nodes in the VANET. Within this model, fog assumes a central position in event validation, leveraging information from mobile nodes (i.e., vehicles) and IoT devices to identify and form confidence values for events and then aggregating the body of evidence to decide on the truth of the event.

Vehicles in the zone of a reported event form event confidence values by aggregating all reports received pertaining to the event (Vehicle Confidence Module). Fog collects information from its nodes and, once a threshold is met, aggregates all available evidence to form a decision (Fog Confidence Module). This decision is then mapped to appropriate actions which are conveyed to the vehicles within the fog server's jurisdiction (Fog Action Module).

A key benefit of positioning fog at the center of the decision-making process is the coherence of actions across all vehicles within the zone. By stipulating a single decision to all vehicles in its zone, fog avoids the disordered vehicle behaviour potentially present in models where vehicles make independent, unilateral decisions. Under fog decision-making, vehicles see the actions of other vehicles as predictable and less alarming. At the vehicle confidence level, the proposed model builds on the trust aggregation methods of [88] by accounting for hop count, the number of received reports, and the possibility of multiple report generators. Fog decision-making employs the Dempster-Shafer rule for combinations [87], extended to incorporate importance factors [86] so that more important evidence carries greater weight.

We have not explicitly incorporated reports from IoT devices into the event validation process with this model, nor does fog count reports from other fog nodes as evidence in decision-making. This information, however, may be invaluable in the event validation. Further, our simulation lacks resolution in parameters such as Vehicle Class Integrity, in which only two vehicle roles are considered (HR and LR and assumes previously determined detection abilities (i.e., ξ). Future work should include a greater spectrum of vehicle roles and consider how sensor/driver detection capabilities are quantified.

6.2 Future Work

6.2.1 Development of RTEAM

In our RTEAM model, the potential consequences of decision-making errors, portrayed by the error intensity curves, are based simply on the number of affected vehicles and proximity of the vehicle to the (purported) event. The concept of error intensity, which is the expected severity of the consequence if an error is committed, is undeveloped in two major ways. First, a more exhaustive set of factors (beyond event type and proximity) contributing to error intensity should be established. These may include variables such as weather, traffic density, and safety features, etc. With more variables, the intensity curves are more aptly described as intensity contours (in *n* dimensions). Second, having identified a reasonable set of such variables, a sizable challenge remains to define error intensity as a function of these variables. Suppose we seek to define I_α as a function of variables X, Y, and Z. X may be the vehicle's distance from the event, Y the type of event, and Z the vehicle's speed. Note that X, Y, and Z are variables that are measured in the VANET (for each vehicle); thus, they are known at the time of decision. Given X=x, Y=y, and Z=z, the expected severity of consequence of an error for a given vehicle will still be uncertain since x, y, and z do not constitute all variables relevant to the outcome. There is an innumerable set of variables, $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)$, which cannot be practically accounted for (or conceived of) but affect the severity of consequence. The severity of consequence, conditional on vehicle having committed a *Type I* error (*ERR*1) and on X=x, Y=y, and Z=z is therefore

$$g(\theta|x, y, z, ERR1)$$

The error intensity, I_α, as a function of the measured variables (x, y, z) is defined as the expected severity of consequence:

$$I_\alpha(x, y, z) = E[g(\theta|X, Y, Z, ERR1)] = \int_{-\infty}^{\infty} g(\theta|x, y, z, ERR1) f(\theta|x, y, z, ERR1)\, d\theta$$

where $f(\theta|x, y, z, ERR1)$ is the joint density function of $\theta_1, \theta_2, \ldots \theta_n$, conditional on X=x, Y=y, Z=z, and *ERR*1. Although $I_\alpha(x, y, z)$ will not be obtained analytically, a practical way (in the future) may be to approximate the function based on empirical evidence collected in the VANET. For every decision error (*ERR*1 in this example) made, the consequence for each vehicle is recorded, along with the values of each variable (x, y, z) for each vehicle. As more data are collected, more outcomes can be mapped to smaller and smaller regions of the (measured) variable space (x, y, z in this example). An uncertain contour develops (due to a spectrum of outcomes even within small regions of the variable space because of the unmeasured variables, $\boldsymbol{\theta}$) but when collapsed to the mean values in each region, provides an estimate of the expected consequence severity, I_α, as a function of x, y, and z. The same approach can also be used to estimate I_β. Lacking empirical data, we have resorted in our

model to estimate the error intensities as curves (in 2 dimensions) based on judgement. Refinement to this concept of error intensity contours and the development of methods for characterizing the contours is a prospective research topic.

As mentioned in Section 6.1.1, more holistic estimates for likelihoods (of wrong decisions) may be made by incorporating information from IoT devices, other fog servers, etc., in addition to trust as defined in our RTEAM risk estimation module. The risk assessment may be further enhanced by considering the cluster vulnerability. Finally, research building on RTEAM may be expanded by implementing more comprehensive simulations of diverse scenarios, and through more in-depth comparison of performance against existing models.

6.2.2 Smart Employment in VANETs and Robustness of TER

➢ Smart Employment in VANETs based on TER

By applying TER, we can improve the evaluation process. Furthermore, we believe that if the method of evaluating trust could reflect the performance level according to certain criteria as well, this will be a significant step towards the smart employment of vehicles on the road. The infrastructures and vehicles can employ/ rely on not only the most trusted node but also the most competent one at performing a specific type of task. In the following points, we list some possible tasks and present the role of fog in each task, where fog nodes could be collectors, monitors, or evaluators. We label the task, describe the task, explain the possible performance evaluation method, and determine the evaluator. Here, we define seven possible tasks as follows:

- ***Cluster Leadership (Cluster Head)***

A cluster head (CH) is a node chosen either randomly or based on criteria (e.g., the highest trust value) to lead a group of vehicles. CH is usually responsible for cluster operations, such as managing the cluster membership and making decisions for situations requiring action. In the last decade, several CH selection algorithms have been proposed and were reviewed in [63]. The CH is evaluated by the cluster members who join the cluster based on metrics such as flexibility and fast response (any cluster member can evaluate the CH regardless of its trust level).

The cluster members send their evaluation to the nearest fog node, which then filters the evaluations (i.e., ignores malicious and bad-mouthing evaluations). Fog nodes can also evaluate the CH performance based on the cluster size (i.e., the number of cluster members managed), the number of successful decisions, and performance in detecting malicious nodes/messages.

- ***Giving Recommendations (Recommender)***

In VANETs, when CH, fog, or any member wants to know the trust level of a vehicle with whom they lack previous direct experience with, they may ask for advice from the neighbouring vehicle, who may have established direct experience regarding the node of interest. A recommender is a node (e.g., node A) who provides its trust evaluation of another node whom it has direct experience with (e.g., node B) to a third node (e.g., node C) who does not have direct experience with the node in question (i.e., A has interacted with B but C has not interacted with B; thus node A makes a recommendation to C whether to interact with/trust or do nor trust B). Any node that receives a recommendation from its neighbour evaluates the referrer (i.e., the node which sends the recommendation) based on the reliability of its recommendation (e.g., the reliable recommendation should reflect the real trust level of the node). The evaluation of the referrer is sent to fog for aggregation (i.e., fog is a collector).

- ***Monitoring nodes (Monitor)***

In clustered network, once a cluster is established, all/some members collaborate to monitor each other (i.e., watchdogs). The purpose of monitoring is to detect any malicious or misbehaving node and report its action. Similar to [80], the CH assigns monitors for each cluster member such that the monitor's trust level should be higher than the node it monitors or at least the role of the monitor is higher than the node that it needs to be monitored (i.e., the monitor is a governmental vehicle). The CH evaluates each monitor's reports, as well as the monitors themselves, based on the reliability of their reports and the behaviours. The CH then sends the evaluation to the nearest fog. For example, if a monitor reports a misbehaving node to the CH, the CH then checks the reliability of the monitor's report. If the report is reliable, then the CH suspends the malicious node, evaluates the monitor, and informs the closest fog. Like the clustered network, the monitor in the non-clustered network has to send its observation in case of detecting any suspension behaviour or unreliable report to the closest fog node. Fog nodes monitor the watchdogs and evaluate their performance based on the sensitivity of detecting the misbehaving nodes (i.e., fog is a monitor and evaluator).

- ***Collaboration (Collaborator)***

A collaborator is a node who provides a service or reports with or without being requested to do so (e.g., reporting a personal observation about the node's vehicle such as "Your tires need to be changed"). The collaboration could be vehicle to vehicle or vehicle to fog (i.e., V2V or V2F). If a node receives a truthful and useful report from its neighbour, it evaluates the node positively as a collaborator. The evaluation is then sent to the CH or fog. This type of task-based trust encourages peers to provide their personal observations.

- ***Event Reporting (Reporter)***

We use the same category described in [47] to be part of our task-based model (this point with the two followed points are from [47]) to evaluate the experience of the node based on its relationship to the event. An event reporter is a node that reports the occurrence of an event that it was involved in (e.g., the node vehicle itself is crashed) to its neighbours. When a vehicle receives a report about an event, it evaluates the reliability of the event report according to the available information about the event and the reporters. If the event is true, the reporter is positively trusted. Otherwise, the reporter is evaluated as untrusted. Later, this evaluation is collected by fog.

- ***Event Observing (Witness)***

A witness is an event observer who is within one hop from the event reporter and can observe the event unobstructed. Although an event such as an accident or congestion is detected by sensors on a vehicle, but this is not always the case, and we still rely on the biological senses of the human driver (i.e., witness). Witnesses send observations to the CH, neighbours, or fog, who then makes a decision based on the reports received for the event. If the witness reports a fake event, the witness is evaluated as untrusted; otherwise, the witness is trusted.

- ***Event Participant (Propagator)***

A propagator is a node within two hops or more from the event reporter. The node cannot directly observe the event but it propagates the reports that are received about the event. Note that the event propagator can only relay the message as received (i.e., a propagator cannot modify the message). Any node (e.g., CH or vehicle) evaluates the node based on the reliability of its report (i.e., if the event is reliable, the node is positively evaluated; otherwise, the node is evaluated as distrusted).

Note that the nodes that report unreliable information will be penalized by the fog (i.e., the fog node will decrease its trust values as a reporter, witness, or propagator, depending on the task type). For false events, the event reporter will receive the highest penalty since it is the generator of the untruthful report, while the lowest penalty will go to the propagator since it simply forwards the received message. The penalty factors are ranked as follows,

$$P_{Propagator} < P_{Witness} < P_{Reporter}$$

➢ Robustness of TER

As identified in Section 6.1.2, TER incurs a tradeoff in trust evaluation between the resolution of task types and the amount of data available per task. In cases where data for a specific task (e.g., task A) is sparce, is utility in considering a node's trust scores for other task types (e.g., task B) where the trust scores are based on more abundant data – even if these task types differ from the one at hand but it is under the same category of the other task (task A). Thus, a more robust version of TER may be

one where trust evaluation method is ordinarily based on task type but may also resort to general trust scores in the scarcity of specific trust data. Strategies for adopting and optimizing this type of flexibility in the VANET are recommended for future work. Also, the concept of TER can be extended to include the task evaluation not only based on its type but also the task repetition time and the task affection on the network.

6.2.3 Development of FEVM

In our model, malicious vehicles are penalized with low reputation scores (based on historical results), whereas honest and reliable vehicles garner high trust scores. However, an issue exists with a vehicle that has a high TER for event validation becomes a malicious vehicle. This represents a possibility for a malicious vehicle with a high reputation score to cause havoc in the VANET. Strategies to guard against this vulnerability should be considered in future work.

Future work should also aim to incorporate reports from IoT devices and other fog servers into the vehicle confidence and fog confidence modules, resulting in a more informed and holistic view of reality.

Strategies to permit vehicles to make decisions in circumstances where time is constrained or fog is not responding within a reasonable time should be pursued. Regarding the fog time threshold, future work should also consider expressing this threshold as a function of variables, such as traffic density, weather, etc. Fog does not require reports strictly from vehicles to implement the confidence and action modules. Another consideration for fog time threshold is to have a threshold for revision of the event probability, m(E). Once fog has made a decision, it will continue to receive further evidence (reports). At some point, it is reasonable for fog to reassess the evidence. This concept can also be applied to the Vehicle Confidence Level Module where vehicles will continue to receive additional reports after they have sent their report to fog. At some point, they may transmit an updated confidence level in a report to fog. These processes are shown in the flowchart of Figure 6.1.

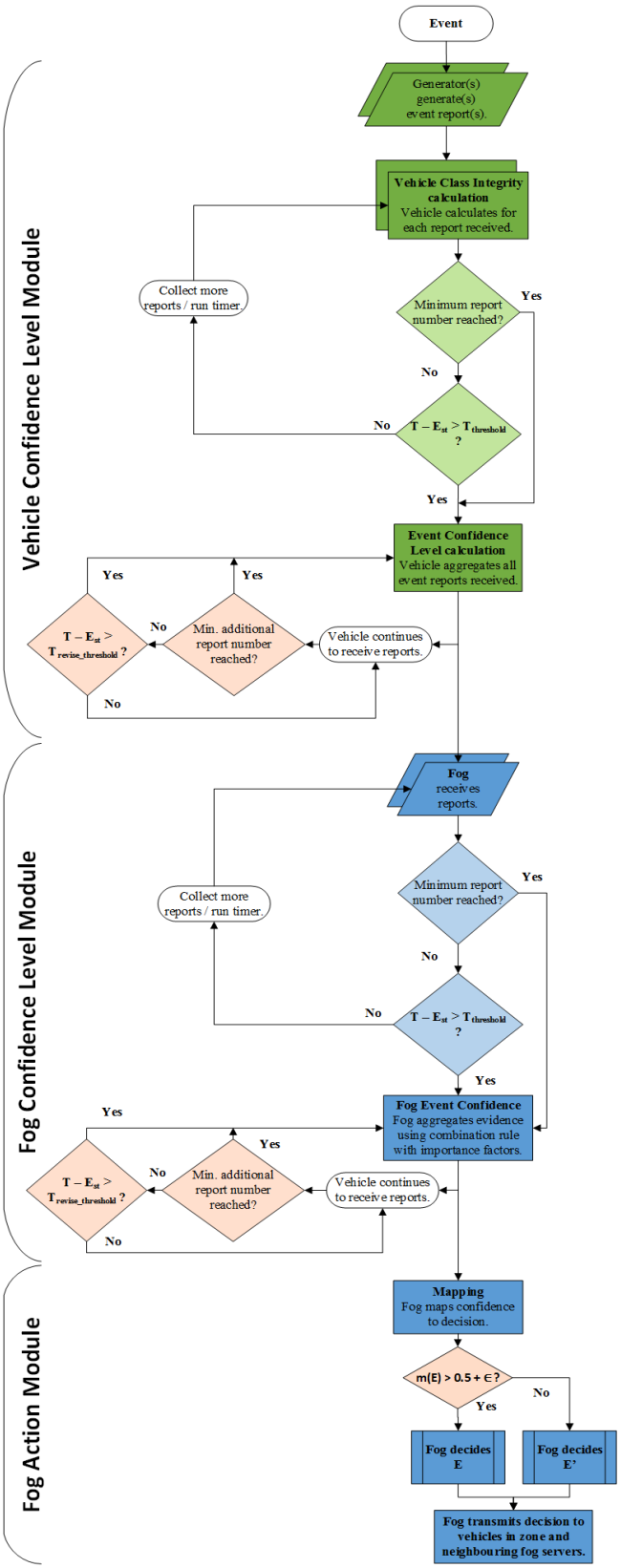


Figure 6.1: Developed FEVM flowchart.

6.2.4 Expansion of Fog Services

We emphasize the synergistic cooperation of fog nodes for evaluating trust in a VANET environment. Fog can also be used to provide the following services:

1. Event Detection

Evaluating the trustworthiness of an event report according to the available data about the sender(s) (i.e., direct experience trust value) and the reported event (e.g., location) is a critical but difficult task due to the nature of VANETs (i.e., high mobility and short connection time). The reliability of the evaluation process is hindered when data is scarce or the number of malicious agents in the network is high. Fog nodes must employ as many resources as feasible. In the edge layer, this may include sensors, radars, and cameras (i.e., IoT devices), in addition to vehicles. Fog can also access resources in the main and intermediate layers such as the cloud, RSUs, and neighbouring fog nodes. Collectively, these resources endow fog with a more accurate and robust interpretation of the daily events that occur within its zone. Partial or full reliance may be placed on fog nodes for event detection, as described below.

a) Partial Reliance

Let N be the number of reports received by fog node k from R available road resources within its zone that claim the occurrence of an event i. Fog k can compute its confidence c_i in the event i as

$$c_i = N/R,$$

where $c_i \in [0, 1]$. If $c_i \geq 0.5$ (i.e., more than half of the road resources confirm the event's existence), fog sends a warning message about event i, including information such as the event ID, type, location, stamp time, and expiration time. Furthermore, fog sends its confidence value c_i to all road users (i.e., all designated cluster members if the network is clustered, or the vehicles in the zone if the network is non-clustered) and the neighbouring fog nodes upstream of the event location. Road users use this confidence in their computation to check whether the event is reliable or not and then make the appropriate decision.

b) Full Reliance

In this situation, fog nodes validate event detection and warn the road users about the upcoming events without the need to perform any reliability checks by the road users. It is important to note that assigning more functions to fog nodes reduces the workload borne by road users.

2. Cluster Head (CH) Selection

In the last decade, cluster head selection has been extensively researched. As mentioned before, numerous papers have been published regarding CH selection. CH selection algorithms vary in the initiation of the election process, the criteria for

selecting the CH, management of cluster membership, and the responsibilities of the CH. In clustered networks, we assume fog nodes will have the option to participate in CH selection. Equipping fog nodes with DTC records of all vehicles passing through their zones enables fog to empirically appraise the vehicles. This elevates fog as a valuable participant in the cluster head selection process. When the cluster head election is initiated, fog is invited or directed to vote. Priority in CH selection is given to the vehicle selected by fog contingent on meeting all other necessary criteria. Incorporating fog in cluster head selection enhances results since fog is a trusted party and has significant background information on the vehicles it appraises for CH (i.e., fog votes for the most trusted and competent vehicle to be CH).

3. Detection/Filtering/Monitoring of Misbehaving nodes

The VANET environment is a multi-node network where nodes communicate directly or indirectly and evaluate each other. Nodes having direct experiences may evaluate each other negatively on unreasonable grounds (e.g., racism, anger). Fog nodes must therefore incorporate effective algorithms for detecting an eliminating malicious report (i.e., badmouthing attacks). These algorithms must effectively screen out biased and unreliable evaluations to improve the quality and reliability of the received evaluations. Authors in [85], reviewed some recently proposed malicious detection techniques in VANETs that can be used by fog nodes.

Fog nodes are distributed along the transportation route and are networked, enabling effective monitoring of the vehicles on the route. Once detected, fog can monitor and suspend a reported misbehaving node. If fog detects a misbehaving node, it may warn the vehicles on the road and as well as the RSU and neighbouring fog nodes. Note that fog requires evaluating generators of malicious reports accordingly.

4. Trust Evaluation

According to [84], fog can compute the trust score of the vehicle on-the-fly without interaction with the cloud and can therefore assume the role of the evaluator in addition to records collector. In our work, fog is the centrepiece of the trust management model, with each fog server constructing its own local reputation system based on local nodes. Unlike vehicles, which are moving, temporary nodes that communicate and evaluate each other, the fog node is a static object that evaluates the moving vehicles according to its observations and metrics.

References

[1] Global Status Report On Road Safety 2018. [online] Available at: https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/ [Accessed 9 April 2020].

[2] How many cars are there in the world?, A. (2018). How many cars are there in the world? Retrieved 2021, from https://www.carsguide.com.au/car-advice/how-many-cars-are-there-in-the-world-70629

[3] Z. Lu, G. Qu and Z. Liu, "A Survey on Recent Advances in Vehicular Network Security, Trust, and Privacy," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp. 760-776, 2019.

[4] N. Lyamin, A. Vinel, M. Jonsson, and B. Bellalta, "Cooperative awareness in VANETs: On ETSI EN 302 637-2 performance." IEEE Transactions on Vehicular Technology, vol. 67, no. 1, pp. 17-28, 2017.

[5] M. Chaqfeh, A. Lakas, and I. Jawhar, "A survey on data dissemination in vehicular ad hoc networks," Vehicular Communications, vol. 1, no. 4, pp. 214-225, 2014.

[6] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "Bars: a blockchain-based anonymous reputation system for trust management in vanets," in Proc. of the 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 98-103, 2018.

[7] 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.

[8] 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.

[9] M. Monir, A. Abdel-Hamid, and M. A. El Aziz, "A categorized trust-based message reporting scheme for VANETs," in Proc. of the International Conf. on Security of Information and Communication Networks, pp. 65-83, 2013.

[10] A. Avizienis, J-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11-33, 2014.

[11] J. H. Cho, A. Swami, and R Chen, “A survey on trust management for mobile ad hoc networks,” IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 562-583, 2011.

[12] Engoulou, Richard Gilles, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero, "VANET security surveys," Computer Communications, vol. 44, pp. 1-13, 2014.

[13] J. Zhang, Q. Zhang, and W. Jia, "VC-MAC: A cooperative MAC protocol in vehicular networks," IEEE Transactions on Vehicular Technology, 58(3), pp.1561-1571, 2008.

[14] P. Dai, K. Liu, L. Feng, Q. Zhuge, V. C. Lee, and S. H. Son, "Adaptive scheduling for real-time and temporal information services in vehicular networks," Transportation Research Part C: Emerging Technologies, vol. 71, pp. 313-332, 2016.

[15] 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. of the IEEE Wireless Communications and Networking Conference (WCNC), pp. 1-7, 2019.

[16] J. Zhang, "Trust management for VANETs: challenges, desired properties and future directions," International Journal of Distributed Systems and Technologies (IJDST), vol. 3, no. 1, pp. 48-62, 2012.

[17] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," Journal of Cloud Computing, 6(1), p.19, 2017.

[18] K. Gu, X. Dong, and W. Jia, "Malicious Node Detection Scheme Based on Correlation of Data and Network Topology in Fog Computing-based VANETs," IEEE Transactions on Cloud Computing, 2020.

[19] Z. Huang, S. Ruj, M. A. Cavenaghi, M. Stojmenovic, and A. Nayak, “A social network approach to trust management in VANETs,” Peer-to-peer Networking and Applications, vol. 7, no. 3, pp. 229-242, 2014.

[20] S. Djahel, R. Doolan, G. Muntean and J. Murphy, "A Communications-Oriented Perspective on Traffic Management Systems for Smart Cities: Challenges and Innovative Approaches," IEEE Communications Surveys & Tutorials, vol. 17, no. 1, pp. 125-151, 2015.

[21] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," IEEE Communications Magazine, vol. 46, no. 6, pp.164-171, 2008.

[22] S. S. Tangade and S. S. Manvi, "A survey on attacks, security and trust management solutions in VANETs," in Proc. of the 4th IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-6, 2013.

[23] D. Zelikman and M. Segal, "Reducing Interferences in VANETs," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 3, pp. 1582-1587, 2015.

[24] M. U. Ghazi, M. A. K. Khattak, B. Shabir, A. W. Malik, and M. S. Ramzan, "Emergency Message Dissemination in Vehicular Networks: A Review," IEEE Access, vol. 8, pp. 38606-38621, 2020.

[25] J. Zhang, F-Y. Wang, K. Wang, W-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, 2011.

[26] A. Laouiti, A. Qayyum, and M. N. M. Saad, Vehicular ad-hoc networks for smart cities. Springer Singapore, 2017.

[27] S. Yang, Z. Liu, J. Li, S. Wang, and F. Yang, "Anomaly detection for Internet of Vehicles: A trust management scheme with affinity propagation," Mobile Information Systems, 2016.

[28] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A Survey of Current Solutions and Future Research Opportunities," IEEE Transactions on Intelligent Transportation Systems, 2020.

[29] C. Huang, R. Lu, and K. K. R. Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges," IEEECommunications Magazine, 55(11), pp. 105-111, 2017.

[30] F. Ahmad, A. Adnane, V. N. L. Franqueira, F. Kurugollu, and L. Liu, "Man-In-The-Middle Attacks in Vehicular Ad-Hoc Networks: Evaluating the Impact of Attackers' Strategies," Sensors vol. 18, no. 11: 4040, 2018.

[31] R. C. Mayer, J. H. Davis, and F. D. Schoorman, "An integrative model of organizational trust," Academy of Management Review. 20 (3), pp. 709–734, 1995.

[32] W. Bamberger, "Interpersonal Trust – Attempt of a Definition," Scientific report, Technische Universität München, 2010.

[33] J. Luo, X. Liu, and M. Fan, "A trust model based on fuzzy recommendation for mobile ad-hoc networks," Computer Networks, vol. 53, no. 14, pp. 2396-2407, 2009.

[34] S. Mandala, A. H. Abdullah, A. S. Ismail, H. Haron, M. A. Ngadi, and Y. Coulibaly, "A review of blackhole attack in mobile adhoc network." in Proc. of the 3rd IEEE International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME), pp. 339-344, 2013.

[35] S. Mandala, K. Jenni, M. A. Ngadi, M. Kamat, and Y. Coulibaly, "Quantifying the severity of blackhole attack in wireless mobile adhoc networks," in Proc. of the International Symposium on Security in Computing and Communication, pp. 57-67, 2014.

[36] N. Bißmeyer, S. Mauthofer, K. M. Bayarou, and F. Kargl, "Assessment of node trustworthiness in vanets using data plausibility checks with particle filters," in Proc. of the IEEE Vehicular Networking Conference (VNC), pp. 78-85, 2012.

[37] Y. M. Chen and Y. C. Wei, "A beacon-based trust management system for enhancing user centric location privacy in VANETs," Journal of Communications and Networks, 15(2), pp.153-163, 2013.

[38] F. G. Mármol and G. M. Pérez, "TRIP, a trust and reputation infrastructure-based proposal for vehicular ad hoc networks," Journal of Network and Computer Applications, vol. 35, no. 3, pp. 934-941, 2012.

[39] O. Abumansoor and A. Boukerche, "Towards a Secure Trust Model for Vehicular Ad Hoc Networks Services," in Proc. of the IEEE Global Telecommunications Conference, pp. 1-5, 2011.

[40] S. A. Soleymani, A. H. Abdullah, W. H. Hassan, M. H. Anisi, S. Goudarzi, M. A. R. Baee, and S. Mandala, "Trust management in vehicular ad hoc network: a systematic review," EURASIP Journal on Wireless Communications and Networking, no. 1(146), pp. 1-22, 2015.

[41] U. Khan, S. Agrawal, and S. Silakari, "Detection of malicious nodes (dmn) in vehicular ad-hoc networks," Procedia Computer Science, vol. 46, no. 9, pp. 965-972, 2015.

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

[43] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, “Towards expanded trust management for agents in vehicular ad-hoc networks,” International Journal of Computational Intelligence: Theory and Practice (IJCITP), vol. 5, no. 1, pp. 3-15, 2010.

[44] 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," Computer Communications, vol. 93, pp. 68-83, 2016.

[45] H. A. Khattak, H. Farman, B. Jan, and I. Ud Din, “Toward Integrating Vehicular Clouds with IoT for Smart City Services,” IEEE Network, vol. 33, no. 2, pp. 65-71, 2019.

[46] Q. Ding, X. Li, M. Jiang, and X. Zhou, "Reputation management in vehicular ad hoc networks," in Proc. of the IEEE International Conference on Multimedia Technology, pp. 1-5, 2010.

[47] Q. Ding, X. Li, M. Jiang, and X. Zhou, “A novel reputation management framework for vehicular ad hoc networks,” International Journal of Multimedia Technology, vol. 3, no. 2, pp. 62-66, 2013.

[48] S. Gurung, D. Lin, A. Squicciarini, and E. Bertino, "Information-oriented trustworthiness evaluation in vehicular ad-hoc networks," in Proc. of the International Conf. on Network and System Security, pp. 94-108, 2013.

[49] D. B. Rawat, Y. Gongjun, B. B. Bista, and M. C. Weigle, "Trust On the Security of Wireless Vehicular Ad-hoc Networking," Ad Hoc & Sensor Wireless Networks, vol. 24, no. 3-4, pp. 283-305, 2015.

[50] R. A. Shaikh and A. S. Alzahrani, "Intrusion-aware trust model for vehicular ad hoc networks," Security and Communication Networks, vol. 7, no. 11, pp. 1652-1669, 2014.

[51] S. A. Soleymani, A. H. Abdullah, M. Zareei, M. H. Anisi, C. Vargas-Rosales, M. H. Khan, 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.

[52] D. Ren, S. Du, and H. Zhu, "A novel attack tree based risk assessment approach for location privacy preservation in the VANETs," in Proc. of the IEEE International Conference on Communications (ICC), pp. 1-5, 2011.

[53] H. K. Kong, T. S. Kim, and M. K. Hong, "A Security Risk Assessment Framework for Smart Car," in Proc. of the 10th IEEE International Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 102-108, 2016.

[54] F. Ahmad and A. Adnane, "A novel context-based risk assessment approach in vehicular networks," in Proc. of the 30th International Conf. on Advanced Information Networking and Applications Workshops (WAINA), 2016.

[55] E. Fitzgerald and B. Landfeldt, "Increasing road traffic throughput through dynamic traffic accident risk mitigation," Journal of Transportation Technologies, vol. 5, no. 04, p. 223, 2015.

[56] E. Fitzgerald and B. Landfeldt, "A system for coupled road traffic utility maximisation and risk management using VANET," in Proc. of the15th International IEEE Conference on Intelligent Transportation Systems, pp. 1880-1887, 2012.

[57] R. A. Shaikh, "Fuzzy risk-based decision method for vehicular ad hoc networks," International Journal of Advanced Computer Science and Applications, vol. 7, no. 9, pp. 54-62, 2016.

[58] 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. of Vehicular Ad-Hoc Networks for Smart Cities, pp. 71-83, 2017.

[59] T. Biswas, A. Sanzgiri and S. Upadhyaya, "Building Long Term Trust in Vehicular Networks," in Proc of the 83rd IEEE Vehicular Technology Conference (VTC Spring), pp. 1-5, 2016.

[60] H. A. Khattak, S. U. Islam, I. U. Din, and M. Guizani, "Integrating fog computing with VANETs: A consumer perspective," IEEE Communications Standards Magazine, 3(1), pp.19-25, 2019.

[61] G. Hampel, K. L. Clarkson, J. D. Hobby, and P. A. Polakos, "The tradeoff between coverage and capacity in dynamic optimization of 3G cellular networks," in Proc. of the 58th IEEE Vehicular Technology Conference (VTC 2003-Fall), vol. 2, pp. 927-932, 2003.

[62] D. Huang, T. Xing, and H. Wu, "Mobile cloud computing service models: a user-centric approach," IEEE Network, vol. 27, no. 5, pp. 6-11, 2013.

[63] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," IEEE Transactions on Vehicular Technology, 65(6), pp. 3860-3873, 2016.

[64] N. Fernando, S.W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," Future generation computer systems, 29(1), pp. 84-106, 2013.

[65] A. Tuli, N. Hasteer, M. Sharma, and A. Bansal, "Exploring challenges in mobile cloud computing: An overview," in Proc. of the 4th International Conference on the Next Generation Information Technology Summit. pp. 9-13, 2013.

[66] H. Atlam, R. Walters, and G. Wills, "Fog computing and the internet of things: a review," Big Data and Cognitive Computing, vol. 2, no. 2, p.10, 2018.

[67] F. Bonomi, "Connected vehicles, the internet of things, and fog computing," in Proc. of the 8th ACM International Workshop on Vehicular Internetworking, pp. 13-15, 2011.

[68] B. Qin, J. Cai, Y. Luo, F. Zheng, J. Zhang, and Q. Luo, "Research and application of intelligent internet of vehicles model based on fog computing," in Proc. of the 3rd IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 1777-1783, 2019.

[69] A. Ullah, S. Yaqoob, M. Imran, and H. Ning, "Emergency message dissemination schemes based on congestion avoidance in VANET and vehicular FoG computing," IEEE Access 7: 1570-1585, 2018.

[70] W. Zhang, Z. Zhang and H. Chao, "Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management," IEEE Communications Magazine, vol. 55, no. 12, pp. 60-67, 2017, doi: 10.1109/MCOM.2017.1700208.

[71] A. Ullah, X. Yao, S. Shaheen and H. Ning, "Advances in Position Based Routing Towards ITS Enabled FoG-Oriented VANET–A Survey," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 2, pp. 828-840, Feb. 2020, doi: 10.1109/TITS.2019.2893067.

[72] C. English, S. Terzis, and W. Wagealla, "Engineering trust based collaborations in a global computing environment," in Proc. of the International Conf. on Trust Management, pp. 120-134, 2004.

[73] X. Zhang, C. Lyu, Z. Shi, D. Li, N. N. Xiong, and C-H. Chi, "Reliable multiservice delivery in fog-enabled VANETs: Integrated misbehavior detection and tolerance," IEEE Access 7: 95762-95778, 2019.

[74] R. Al-ani, B. Zhou, Q. Shi and A. Sagheer, "A Survey on Secure Safety Applications in VANET," in Proc. of the IEEE 20th International Conf. on High Performance Computing and Communications; IEEE 16th International Conf. on Smart City; IEEE 4th International Conf. on Data Science and Systems (HPCC/SmartCity/DSS), pp. 1485-1490, 2018.

[75] J. Zhang, L. Huang, H. Xu, M. Xiao, and W. Guo, "An incremental bp neural network based spurious message filter for vanet," in Proc. of the IEEE International Conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 360-367, 2012.

[76] J. Zhang, X. Tan, X. Wang, A. Yan, and Z. Qin, ‘‘T2FA: Transparent two factor authentication,’’ IEEE Access, vol. 6, pp. 32677–32686, 2018.

[77] D. Zhang, F. R. Yu, and R. Yang, "Blockchain-Based Distributed Software-Defined Vehicular Networks: A Dueling Deep ${Q} $-Learning Approach," IEEE Transactions on Cognitive Communications and Networking, 5(4), pp.1086-1100, 2019.

[78] G. Stoneburner, A. Y. Goguen, and A. Feringa, “Risk management guide for information technology systems,” NIST Special Publication, 2002.

[79] G. Parmigiani and L. Inoue, “Decision theory–principles and approaches,” Chichester, John Wiley & Sons Ltd, 2009.

[80] R. Sugumar, A. Rengarajan, and C. Jayakumar, “Trust-based authentication technique for cluster-based vehicular ad hoc networks,” Wireless Networks, vol. 24, no. 2, pp. 373-382, 2018.

[81] R. Azizi and G. Oz, "Performance evaluation of data dissemination in real-world wireless ad hoc networks," in Proc.of the International Conf. on Communications and Information Technology (ICCIT), 2011.

[82] T. Thenmozhi, and R. M. Somasundaram. "Towards modelling a trusted and secured centralised reputation system for VANET’s." in Proc. of the International Conf. on Soft Computing Systems, pp. 675-688. Springer, 2016.

[83] R. Iqbal, TA. Butt, M. Afzaal, and K. Salah. "Trust management in social internet of vehicles: factors, challenges, blockchain, and fog solutions." International Journal of Distributed Sensor Networks 15, no. 1, 2019.

[84] P. Hu, S. Dhelim, H. Ning, and T. Qiu. "Survey on fog computing: architecture, key technologies, applications and open issues." Journal of network and computer applications 98, pp. 27-42, 2017.

[85] M. Arshad, Z. Ullah, N. Ahmed, M. Khalid, H.Criuckshank, & Y. Cao "A survey of local/cooperative-based malicious information detection techniques in VANETs". EURASIP Journal on Wireless Communications and Networking, pp.1-17, 2018.

[86] Z. Zhao, H. Hu, G. Ahn and R. Wu, "Risk-Aware Mitigation for MANET Routing Attacks," in IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 2, pp. 250-260, 2012.

[87] G. Shafer."A mathematical theory of evidence". Princeton university press, 1976.

[88] C. Chen, J. Zhang, R.Cohen, & P. H. Ho. "A trust-based message propagation and evaluation framework in vanets". In Proceedings of the Int. Conf. on Information Technology Convergence and Services, June, 2010.

[89] Zadeh, L. A., 1984. “Review of Books: A Mathematical Theory of Evidence.” The AI Magazine, pp. 81-83.,