A Dynamic Trust Management Model for Vehicular Ad Hoc Networks

Mehmet Aslan[a,*], Sevil Sen[a]

[a] *WISE Lab., Department of Computer Engineering, Hacettepe University, Ankara, Turkey*

Abstract

Trust management in vehicular ad hoc networks (VANETs) is a challenging dynamic optimization problem due to their decentralized, infrastructureless, and dynamically changing topology. Evolutionary computation (EC) algorithms are good candidates for solving dynamic optimization problems (DOPs), since they are inspired from the biological evolution that is occurred as a result of changes in the environment. In this study, we explore the use of genetic programming (GP) algorithm and evolutionary dynamic optimization (EDO) techniques to build a dynamic trust management model for VANETs. The proposed dynamic trust management model properly evaluates the trustworthiness of vehicles and their messages in the simulation of experimental scenarios including bogus information attacks. The simulation results show that the evolved trust calculation formula prevents the propagation of bogus messages over VANETs successfully and the dynamic trust management model detects changes in the problem and reacts to them in a timely manner. The best evolved formula achieves 89.38% Matthews Correlation Coefficient (MCC), 91.81% detection rate (DR), and 1.01% false positive rate (FPR), when $\approx$ 5% of the network traffic is malicious. The formula obtains 87.33% MCC, 92.01% DR, and 4.8% FPR when $\approx$ 40% of the network traffic is malicious, demonstrating its robustness to increasing malicious messages. The proposed model is also run on a real-world traffic

*Corresponding author

Email addresses: mehmetaslan@cs.hacettepe.edu.tr (Mehmet Aslan), ssen@cs.hacettepe.edu.tr (Sevil Sen)

URL: https://wise.cs.hacettepe.edu.tr/ (Mehmet Aslan), https://wise.cs.hacettepe.edu.tr/ (Sevil Sen)

model and obtains high MCC and low FPR values. To the best of our knowledge, this is the first application of EC and EDO techniques that generate a trust formula automatically for dynamic trust management in VANETs.

Keywords: Vehicular Ad Hoc Networks, Security, Trust Management, Evolutionary Computation, Genetic Programming, Evolutionary Dynamic Optimization

---

1. Introduction

Vehicles have been equipped with various smart modules to ensure safer, efficient, and reliable road transportation in recent years. These modules are forming intelligent transportation systems (ITS) that cover different aspects of transportation and traffic management. One of these systems is vehicular ad hoc networks (VANETs), which are a form of mobile ad hoc networks (MANETs) in the vehicle domain. They are mobile, decentralized, infrastructureless wireless networks that provide vehicles to communicate with other vehicles on the road for sharing information about safety warnings, road status, and advertising services.

Inherent characteristics of VANETs bring some security challenges. Since they are infrastructureless and decentralized, vehicles can enter to and exit from the network without any control due to the lack of a central management unit or an access point. This makes VANETs vulnerable to several attacks such as bogus information [1], in which attackers modify messages or forge fake messages into the network. Vehicles must distinguish such false messages in order to achieve a reliable communication, hence to maintain the traffic safety and efficiency on the road. In the literature, trust management models are widely proposed as a solution to such attacks. Decentralized, self-organized, autonomous, and highly dynamic topology of ad hoc networks makes the trust management an optimization problem.

Besides its dynamic topology, other dynamicities in VANETs can make the trust management problem harder. The vehicle density of the traffic can change from time to time, such as it can increase at rush hours in urban areas and decrease after a while, this causes difficulties to the solution of the trust management problem because it must perform well in all situations. Similarly, the density of events can also change dynamically at different times. Events are the situations in the traffic which vehicles share information with other vehicles on the road. While vehicles send messages about stationary

events such as services on the road, they could send additional messages about critical events occuring on the road such as traffic accidents, road maintenance in order to increase the traffic safety. The solution to the trust management problem must handle this safety critical dynamicity.

Proposed solutions for the trust management problem in VANETs might be valid for only a length of time due to changes over time in such a dynamic environment. Such optimization problems that change over time in a dynamic environment are called dynamic optimization problems (DOPs) [2], and an optimization algorithm must be able to not only solve the problem at a time but also detect changes in the problem occurring over time in order to search for a new solution. Nature-inspired optimization algorithms are good candidates for solving DOPs, since they are inspired from biological evolution and natural self-organized systems which are dynamic due to their very nature. Using evolutionary computation (EC) techniques to solve DOPs is named evolutionary dynamic optimization (EDO) [2]. Such EDO techniques have already been employed to solve some problems in ad hoc networks [3]. However, there is a lack of studies on real-world EDO applications, so more real-world DOPs need to be modeled and solved by EDO in order to reduce the gap [2].

In this study, an EDO based dynamic trust management model is proposed to evaluate the trustworthiness of both vehicles and messages sent by these vehicles in VANETs, where attackers send bogus information to the network. The previous studies in the literature generally employ statically defined formulas with a limited set of trust evidences for evaluating data or node trust and change the coefficients of parameters in such formulas to deal with dynamicity. On the other hand, the proposed model generates a formula in order to evaluate trust automatically by taking into account much more trust evidences than the studies in the literature. Genetic programming (GP) is explored to evolve the trust formula and EDO techniques are integrated to detect the change in the problem due to the dynamically changing environment over time. To the best of our knowledge, there is no such study that automatically adapts to the environment for managing trust in VANETs. The contribution of this current study could be summarized as follows:

- The use of EC techniques, specifically genetic programming, is explored to distinguish bogus information from legitimate messages using an automatically generated trust calculation formula rather than a pre-

defined static one. The results show that GP could evolve effective formulas in order to evaluate the trustworthiness of messages sent by vehicles, thus leading to effectively evaluate the trustworthiness of these vehicles.

- The effectiveness of trust evidences are explored to satisfy the requirements of trust management systems in VANETs. Differently from the studies in the literature, a broader set of trust evidences is given to the model and the ones that best represent the network for trust calculation are selected by GP.

- The use of EDO techniques is explored for dynamic trust management in VANETs. The results show that EDO could detect changes in the environment automatically and timely, hence able to adapt to such changes quickly.

- To the best of our knowledge, this is the first study that investigates the use of EC techniques for trust management in VANETs. Moreover, it is the first approach that solves the problem from the DOP point of view.

- The model is run on a real-world traffic model to reduce the gap between the EDO research and real-world DOPs.

The rest of this paper is organized as follows: The related studies on trust management systems in VANETs, the use of evolutionary computation techniques in ad hoc networks, and the applications of evolutionary dynamic optimization algorithms in the literature are summarized in Section 2. The network environment used in this study is introduced in Section 3. The proposed dynamic trust management method is given in detail in Section 4. Section 5 presents the experimental settings, scenarios regarding attacks, and dynamic changes employed in the network simulations. Section 6 presents and discusses the experimental results. Section 7 discusses a case study using the proposed model on a traffic model taken from the real world. The limitations of the proposed approach and the possible future research directions for the problem are discussed in Section 8. Finally, the summary of the study is given in Section 9.

2. Related Work

The previous studies are divided into three categories based on their main focus and relevance to this research. The proposed trust management systems for VANETs are reviewed in Section 2.1. Section 2.2 presents the proposed solutions based on evolutionary computation for solving problems in ad hoc networks. The use of evolutionary dynamic optimization algorithms in the literature is summarized in Section 2.3.

2.1. Trust Management in VANETs

Many aspects should be taken into account to establish a proper trust-based framework for both VANETs and other ad hoc networks. These aspects, called as trust management components, are defined as properties of trust, trust management properties, trust metrics, and attacks to the trust model in several surveys [4, 5, 6, 7, 8]. Dynamicity, incomplete/partial transitivity, context-dependency are described in [4, 6] and subjectivity and asymmetry are also described in [4] as trust properties. Nonetheless, none of the proposed approaches for VANETs covers all trust properties [4].

In highly dynamic and distributed environments such as VANETs, trust management should be fully decentralized [8]. It is described as one of the most important trust management properties, since a centralized authority cannot be assumed to be existing for trust computation in VANETs [4]. Because of the possibility of interaction with the same vehicle might be low in a fast and dynamic VANET environment, vehicles cannot wait until direct interactions reach a threshold [8]. Another property that should be considered is capturing the dynamicity of VANETs in order to calculate the trust based on the current situation using event/task type, location, and time information [8]. Moreover, the possibility of uncooperative vehicles to enter VANETs freely should also be taken into account in developing a trust management model [4, 8].

Decentralized trust models in VANETs that are based on past interactions and environmental information in order to take the dynamic infrastructure of VANETs into consideration are grouped into three categories: entity-oriented, data-oriented, and hybrid trust models [6, 8]. Entity-oriented trust model is the traditional way for trust computing that is proposed for many ad hoc networks including VANETs and MANETs. It only considers the trustworthiness of nodes in the network and does not compute different trust values for different messages sent from the same node. Calculating only the

trustworthiness of messages sent from nodes without considering the trust values of the nodes themselves is called data-oriented trust model. Hybrid trust models evaluate both trust values. In hybrid trust models, the entity trust value is used as another parameter to evaluate the data trust value in addition to trust evidences, and the entity trust value is later updated according to the calculated data trust value in order to maintain a trust relationship based on past interactions.

Chen and Wei [9] proposed a hybrid trust model to evaluate the trustworthiness of an event message using beacon, event, and reputation trust values of the vehicles in VANETs. It employs both beacon messages and event messages to calculate the trust value and update the reputation trust value of vehicles by using the trust value of the latest event. Event messages are forwarded either to support or to deny opinion according to a trust threshold in this model. They simulate the model with scenarios including both alteration attacks and bogus information attacks and evaluate the model using F_1 measure [10]. However, they only consider a vector of position, velocity, and direction values of a vehicle and similarity between the event location and the estimated location of the vehicle as trust evidences with a threshold for the distance between the receiver and the sender and a threshold for the time delay between the event message time and the current time.

Yao et al. [11] proposed an entity-oriented trust model and a data-oriented trust model, however they did not integrate these. Even though they use the trust value of vehicles in VANETs as a parameter of data-oriented trust model, they do not update the trust value of vehicles using the trust value of data sent from it. They take into account different event types and different vehicle types by assigning weights to them and introduce a weighted version of the successful data forwarding rate using the event weights called malicious tendency. This value and vehicle type are then used to calculate the trust values of vehicles in the entity-oriented trust model. They use the distance between the event position and the sender vehicle's position in addition to the trust value of the sender vehicle, and the difference between the time of event occurrence and the time of event message in order to calculate data trust. They focus on enhancing the security of the routing protocol in the network simulations in which black hole attack and selective forwarding attack scenarios as well as a network scenario without attacks are considered. Three network-based metrics of packet delivery ratio, average path length, and average end-to-end delay are used to evaluate the entity-oriented trust model, and a case analysis involving 3 kinds of data from 10 types of nodes

is made for validation of the proposed data-oriented trust model.

Machine learning-based trust management approaches for securing the communication of vehicles in vehicular networks have been emerging recently [12, 13]. A trust-aware support vector machine-based (SVM) intrusion detection system is proposed to assign a trust value for vehicles and detect malicious behaviours in VANETs [14]. An attribute-weighted K-means clustering algorithm, which is based on direct and indirect trust models, is proposed to identify messages as either true or false [15]. A trust-based deep reinforcement learning (Deep RL) algorithm is proposed to select the most trusted routing path for the communication of connected vehicles [16]. In addition, blockchain based trusted communication systems are proposed to ensure the trustworthiness of vehicles and messages in VANETs [17]. To build a secure intelligent transportation system against unauthorized drivers, another study analyzes and processes drivers' behavior using deep learning techniques, presenting a different perspective on the problem [18].

To sum up, the previous studies that focus on decentralized trust models in VANETs either take into account very limited trust evidences or do not attach much importance to hybrid trust models as shown in Table 1. Our previous work [19] proposes a GP based trust management model for VANETs in order to properly evaluate the trustworthiness of data about events. In this paper, we automatically generate a hybrid trust model that mainly aims to evaluate data trustworthiness by using a broader set of trust evidences gathered from the network. Entity trust values of vehicles are calculated based on the data trust values of messages sent by these vehicles. This study improves our previous work by using extended effective trust evidences. Moreover, it approaches the problem as a DOP and hence employs the EDO technique to solve it.

2.2. Evolutionary Computation Techniques in Ad Hoc Networks

Nature-inspired algorithms developed for solving different problems in ad hoc networks are classified according to their execution mode, information requirements, and executing platform in [20]. Firstly, the algorithms are classified as either online or offline techniques based on the execution time of them, during runtime or beforehand. Secondly, the requirement of information about the network is considered and the algorithms are classified as global knowledge if they need the whole network information and local knowledge if the nodes only use information gathered by themselves.

Lastly, the optimization algorithms that are run on a central unit are classified as a centralized system, and the optimization algorithms that are run on each node of the network locally are classified as a decentralized system. Authors also classified existing studies based on this taxonomy, but they did not mention any research about trust management in ad hoc networks. Most of the bio-inspired algorithms used in ad hoc networks are mainly based on two categories, one is centralized and offline with global knowledge and the other is decentralized and online with local knowledge. The latter is more appropriate for trust management in VANETs as each vehicle must evaluate trust values using only its own local information while moving online on the network.

A recent survey reviews the applications of evolutionary algorithms (EA) that are proposed to solve optimization problems in mobile ad hoc networks in the literature [21]. The survey focuses on MANETs, VANETs, and DTNs (delay tolerant networks) and divided the reviewed studies into five categories: topology management, broadcasting algorithms, routing protocols, mobility models, and data dissemination. It did not mention any work based on trust management in ad hoc networks. Another survey focuses on the applications of evolutionary computation methods for cybersecurity in MANETs and covers EA, swarm intelligence (SI), artificial immune systems (AIS), and evolutionary games (EG) [22]. This survey classifies these algorithms based on the attack types that they counteract and the defense mechanisms that are implemented by them, including node trust and reputation systems. It is shown that most of the proposals in the literature are based on EG [22]. While EC techniques are investigated for intrusion detection in many studies both for wired and wireless networks [23], such as a GP and grammatical evolution (GE) study [24], there is only one application of EA to trust and reputation systems and that is proposed for peer-to-peer networks (P2P) [25]. To sum up, as far as we know, the current study is the first application of evolutionary computation techniques to the trust management problem in VANETs.

Trust management models in the literature mainly aim at detecting malicious/untrusted users. However, the complex and dynamic properties of VANETs make the detection of attacks/attackers is hard. Researchers choose a fixed set of parameters to build a trust management system in previous studies, but this approach can not represent the dynamically changing environment of VANETs because a change in the environment can invalidate the chosen parameters, thus the system starts to make wrong decisions. In

this research, this issue has been addressed by using EC techniques to choose the parameters automatically from a broader set and change them according to the dynamicity. EC algorithms require fewer a priori assumptions about the problem at hand [26]. Furthermore, EC seamlessly lends itself to the integration of human expert knowledge as needed, and the representation of solutions in EC algorithms can be quite flexible [26]. These characteristics of EC are among the main motivations behind using EC in this research.

2.3. Evolutionary Dynamic Optimization Algorithms

Different EDO algorithms are reviewed based on their approaches to take into account the dynamics of optimization problems while proposing a new definition of DOPs to distinguish them from other dynamic/time-dependent problems and to prevent using these terms interchangeably in [2]. They point out that optimization algorithms must track the change of the optimal solution because of the time-varying problem while trying to find the optimal solution of the current problem. They classify existing algorithms into categories according to change detection, diversity management, memory usage, prediction, self-adaptation, and multipopulation. They also discuss that there exist a limited number of studies on real-world applications of EDO.

Most of the existing EDO studies on real-world applications are either using genetic algorithms (GA) for DOPs of different areas in MANETs such as dynamic multicasting [27], dynamic shortest path routing [28], dynamic load balanced clustering [29], dynamic routing [30] or ant colony optimization (ACO) for DOPs of areas other than ad hoc networks such as dynamic vehicle routing [31], extended capacitated arc routing [32], dynamic travelling salesman problem with traffic factors [33]. There does not exist any research based on EDO techniques for the dynamic trust management problem in ad hoc networks, so again as far as we know, this is the first study to take into account the trust management problem as a DOP and try to solve it using EDO algorithms.

3. The Network Model

Since there is no well-accepted standard for VANETs yet, an application layer protocol that the proposed trust model is built on is introduced and explained in this section.

Table 1: Summary of the Related Works

Work	**Focus**	**Domain**	**Model**	**Limitations**
[25]	Trust & Reputation	P2P	GP	N/A
[27]	Multicasting	MANETs	EDO by GA	
[28]	Routing			
[29]	Clustering			
[30]	Routing			
[31]	Routing	Vehicles	EDO by ACO	
[32]	Routing			
[18]	Driver Identification		DL	
[14]	Trust-Aware Intrusion Detection	VANETs	SVM	Lack of data trust
[15]	Trust-Aware Clustering		K-means	Lack of hybrid trust
[16]	Trusted Routing		Deep RL	Lack of data trust
[9]	Trust Management		Hybrid Trust	Limited size of trust evidences
[11]	Trust Management		Entity & Data Trust	Lack of hybrid trust
[19]	Trust Management		GP	Not suitable in dynamicity
Our Work	Dynamic Trust Management		EDO by GP	Defining change rate of operators

3.1. Network Assumptions

Vehicular ad hoc networks are formed by vehicles that participate to, and leave from the network dynamically at any time while moving on the road at different speeds and generally arrive at different destinations. These vehicles encounter other vehicles in the traffic and make communication with them

on the move. They contribute to the network communication by sending their own messages and forwarding messages coming from their neighbours to other vehicles. Vehicles generally communicate with each other for a short period of time, then never see each other again, which makes safely communication harder for such dynamic networks. On the other hand, some vehicles might move regularly to the same or similar destinations on different days. This slightly increases the probability of meeting with the same vehicle, thus making it useful to employ past interactions for establishing more safely communication. Unfortunately, there is no standard communication model for VANETs yet, so researchers have been proposing new communication models. In the following, some assumptions about vehicles to propose a communication model are introduced.

All vehicles have the equipment required to communicate with other vehicles over wireless links and to form a VANET. The system times of all vehicles are assumed to be synchronized by GPS as in [9]. They could send messages about the properties of themselves and events on the road to other vehicles within their communication range. They also could process messages coming from their neighbours, extend them by adding fields to the received messages, and forward the extended message to other vehicles in the network. Vehicles have a unit for calculating the trust levels of other vehicles and their messages using some features collected from both the network and the message. Identities and types of all vehicles are assumed to be controlled and signed by the authorities, thus these information cannot be changed by the vehicles themselves. A fully trustworthy authority uses a public key infrastructure and carries out key management, such as issuing certificates to newly registered vehicles, verification of certificates of vehicles, and revocation of certificates, as assumed in [9, 11].

3.2. Application Messages

Many applications running on VANETs mainly focus on sharing information about events that vehicles come across. Vehicles send application layer messages to others while moving on the road to communicate and improve the safety and efficiency of the traffic. These messages mainly have two types: beacon and event messages.

3.2.1. Beacon Messages

Beacon messages are periodically sent messages without an observation of an event. Vehicles send beacon messages every second to their neighbour

nodes that are in their direct communication range. This message shows that the sender vehicle of it is in the traffic network and moving on the road. The beacon message includes the current position and velocity data of vehicle at the time of sending this message in addition to the unique identifier and type of the vehicle as shown in Table 2.

Table 2: Format of the Beacon Message

Unique Identifier	Vehicle Type	Message Time	Current Position	Current Velocity

3.2.2. Event Messages

Event messages are sent by vehicles only when an event is observed. Events can be considered as situations occurring in traffic or roads that are worth to share information about them, such as traffic accidents, traffic jams, or toll roads. Events that could occur in traffic are categorized into three groups: safety events, efficiency events, and infotainment events. Messages about safety events are the most critical type, since it aims to increase traffic safety in critical events such as traffic accidents, wet/icy roads. Efficiency event messages are used in order to establish an efficient traffic network in the case of events such as traffic congestion, road maintenance, and closed roads. Infotainment event messages carry information about the facilities nearby, such as toll roads, scenic areas, restaurants, parking/petrol stations. An event message includes the event type, event description, and event position data besides the fields that exist in beacon messages as shown in Table 3. However, these messages are triggered only when an event occurs, on the contrary to beacon messages, which are sent periodically. In that way, the data trust value of the event message is calculated without beacon messages being stored.

Table 3: Format of the Event Message

Unique Identifier	Vehicle Type	Message Time	Current Position	Current Velocity
	Event Type	Event Description	Event Position	

3.3. Attack Types

Suitable security solutions are needed for VANETs to overcome the vulnerabilities caused by allowing any vehicle to enter to the network, such as selfish vehicles, misbehaving ones, and malicious vehicles. Selfish vehicles use the network for their own intent. They collect all information from other vehicles but do not send any data or send very limited/insufficient data to them. Their main motivation is using the resources for their own good only and not being helpful for other vehicles in the network. Misbehaving vehicles could have some malfunctioned device or could be captured by an attacker and send false information unintentionally. Malicious vehicles aim to damage the network deliberately and are called attackers.

Malicious vehicles can carry out different types of attacks in any communication layer in order to harm VANETs. Benign vehicles should be aware of that kind of attacks and they must decide whether the received messages from other vehicles are trustable or not. Since different kind of attacks requires different security countermeasures, this study focuses on the bogus information attacks. More specifically, proposing a dynamic trust management model for the following two attack types is the main motivation of this study.

3.3.1. False Information Attack

Malicious vehicles observe events on the road like benign vehicles, but they modify such messages about the events before forwarding them. Before forwarding the message to their neighbours, attackers change the event type of the real event as if a different event exists at the same position. This causes vehicles receive conflicting event messages about the event at the same position. If a vehicle is convinced that the event messages received from the attacker are true, it might begin to classify benign vehicles as attackers.

3.3.2. Fake Message Attack

Malicious vehicles forge fake messages about nonexistent events to their neighbours in this attack scenario. While an existent event message is modified in the false information attack, a new one is created in this attack type. Attackers generate and send fake event messages to gain some advantage on the road. For instance, they could decrease the density of a road by sending fake messages about a nonexistent accident on that road. Such fake messages can easily spread across the network. Because unlike the false information

attack, there are no other messages regarding these fake events to help detect the attack.

4. Dynamic Trust Management

Trust management models are used by researchers in ad hoc networks to ensure secure and reliable communication. In such models, each node assigns a trust degree to each message it receives and/or to each node that the message is received from. A trust formula is used to calculate such trust degrees by using the available information in the network. However, generally manually generated trust formulas have a limited number of features and, hence cover only a little aspect of network. They might not be able to represent the complex properties of VANETs. A trust management model proposed for VANETs should be able to reflect changes in topology and events in the model.

In this study, we investigate the use of evolutionary dynamic optimization techniques in order to generate a dynamic trust management model automatically. Hence, the complex properties of VANETs such as dynamically changing topology and events could be taken into account effectively and efficiently. The model generates a formula for trust calculation using a broader set of features, i.e., trust evidence, than previous studies in the literature. The features represent complex characteristics of such a dynamic environment. The components of the proposed dynamic trust management model are described in the following sections.

4.1. Trust Types

Vehicles assign trust values not only to vehicles but also to the event messages that are sent from these vehicles. These types of trust are called vehicle trust and data trust (in other words, event trust), respectively. A vehicle's trust value represents the trustworthiness of vehicles in VANETs. Its main aim is to find malicious vehicles and exclude them from the network. An event's trust value focuses on detecting bogus messages and preventing them to be distributed into the network. These two types of trust values affect each other in order to achieve a dynamically integrated trust model. A more reliable trust management framework could be established by using these two values together.

4.2. Trust Properties

Trust management systems for ad hoc networks should take into account all five properties of trust, dynamicity, incomplete/partial transitivity, context-dependency, subjectivity, asymmetry, as mentioned in Section 2.1.

Vehicles use only the information that they can gather from the network and they express the value of trust as a continuous variable, thus the dynamicity of trust is represented in this study. Each vehicle calculates a different trust value for each event message even if they are sent from the same vehicle to evaluate the different experience with the vehicle, so a subjective trust is established. A weighted transitivity model is used to transfer the trust information about a vehicle to other vehicles to satisfy the incomplete transitivity property of trust. A trust value is calculated only when a vehicle receives an event message, so two vehicles which are communicated with each other do not have the same trust value for each other. In addition, there exist different types of vehicles in this network model that affect directly on their trust values, thus these bring an asymmetric trust. The two different trust types, vehicle trust and data trust, provide context-dependent trust values between two vehicles.

4.3. Trust Evidences

Each term in the trust formula expression is called trust evidence and they represent the features of the network, vehicles, and messages. Each vehicle in the network gathers items of evidence about the network by using both beacon and event messages. The values of items of trust evidence that are used in this study are normalized to $[0, 1]$. Table 4 shows the trust evidence set and Table 5 shows the notations used in the proposed dynamic trust management model, which is described in detail below.

4.3.1. Neighbourhood

Vehicles calculate the current neighbourhood density as the ratio of the number of current neighbours to the number of maximum neighbours encountered up to this time. Number of newly added neighbours and removed neighbours since the delivery of the last event message is monitored by vehicles using beacon messages. The percentages of these values are calculated using the same number of maximum neighbours. The neighbourhood density of the vehicle A, ND_A is defined as in Eq. 1, the percentage of added neighbours of the vehicle A, AP_A is defined as in Eq. 2, and the percentage

Table 4: Trust Evidence Set

Abbr.	**Trust Evidence**
ND	Neighbourhood density
AP	Percentage of added neighbours
RP	Percentage of removed neighbours
EP	Proximity of the receiver vehicle to the event
VP	Proximity of the receiver vehicle to the sender vehicle
SP	Proximity of the sender vehicle to the event
TP	Proximity of the event time to the current time
W_V	Weight of the vehicle
W_E	Weight of the event
PE	Percentage of vehicles sending the same event
PT	Percentage of vehicles sending the same event type
VT	Trust value of the source vehicle
ET	Trust value of the event message
VW	Average weight of the vehicles sending the same event
EW	Average weight of the events at the same location
TV	Average weighted trust value of the source vehicle
TE	Average weighted trust value of the event message
MP	Percentage of malicious messages sent from the vehicle

of removed neighbours of the vehicle A, RP_A is defined as in Eq. 3:

$$ND_A = NN_A \ / \ MN_A \tag{1}$$

$$AP_A = AN_A \ / \ MN_A \tag{2}$$

$$RP_A = RN_A \ / \ MN_A \tag{3}$$

4.3.2. Proximity

Position and time proximity values are important factors in order to decide whether the trust value of an event message and its sender should be

calculated or not. Vehicles calculate three different position proximity values using its own position, position of the received event, and position of the sender vehicle. They also calculate the proximity of the event time to the current time. Some messages are not taken into account for the calculation of trust value when their proximity values exceed the maximum allowed distance and time values. The proximity of the receiver vehicle R to the event X EP_R^X is defined as in Eq. 4, the proximity of receiver vehicle R to the sender vehicle S VP_R^S is defined as in Eq. 5, the proximity of the sender vehicle S to the event X SP_S^X is defined as in Eq. 6 and the proximity of event time X to current time TP_X is defined as in Eq. 7:

$$EP_R^X = (MD \ - \ ED_R^X) \ / \ MD \tag{4}$$

$$VP_R^S = (MD \ - \ VD_R^S) \ / \ MD \tag{5}$$

$$SP_S^X = (MD \ - \ ED_S^X) \ / \ MD \tag{6}$$

$$TP_X = (MT - (T - GT_X)) \ / \ MT \tag{7}$$

4.3.3. Vehicle Type

Vehicles in VANETs have different roles and objectives on traffic based on their types, which are divided into three groups: police cars, public service vehicles, and ordinary automobiles. Vehicle types usually indicate the trustworthiness of vehicles to some extent. Police cars are responsible for controlling the traffic and providing road safety, therefore they are the most trustworthy vehicles in the network. They are considered as vehicles with high trust level in the proposed trust model. Public service vehicles such as ambulances, buses, and engineering vehicles are usually on duty for ensuring either road safety or efficiency, thus they are considered as vehicles with medium trust level. Ordinary automobiles such as private cars, taxis are considered as low level vehicles from the trust point of view, since their contribution to road safety is generally lower than others. To use this knowledge in trust calculations, a trust evidence called vehicle weight $W_V(x)$ is defined as in Eq. 8:

$$W_V(x) = \begin{cases} 1.0, & \textit{when } x \textit{ is a police car} \\ 0.7, & \textit{when } x \textit{ is a public service vehicle} \\ 0.5, & \textit{when } x \textit{ is an ordinary automobile} \end{cases} \tag{8}$$

Table 5: Notations

Notation	Definition
NN_A	number of neighbours of vehicle A
MN_A	maximum number of neighbours of vehicle A
AN_A	added number of neighbours of vehicle A
RN_A	removed number of neighbours of vehicle A
ED_R^X	distance of receiver vehicle R to the event X
VD_R^S	distance of receiver vehicle R to the sender vehicle S
ED_S^X	distance of sender vehicle S to the event X
MD	maximum allowed distance
T	current time
GT_X	generation time of the event message X
MT	maximum allowed event time
W_V^A	weight of the vehicle A
W_E^X	weight of the event X
VT_R^S	trust value of vehicle S calculated by vehicle R
ET_R^X	trust value of event X calculated by vehicle R
EN_A^X	the number of vehicles sending the same event X
TN_A^X	the number of vehicles sending the same event type X
TT	the threshold for classifying an event message
CT_R^S	current trust value of vehicle S calculated by vehicle R
T_R^S	new trust value of vehicle S calculated by vehicle R

4.3.4. Event Type

Events have different impacts on traffic and road safety, thus requiring different trustworthiness levels. The most important event type is clearly safety events as described in Section 3.2.2. Vehicles in VANETs pay attention to the importance levels of events to maintain road safety. This information is represented with a trust evidence called event weight $W_E(x)$ as defined in

Eq. 9:

$$W_E(x) = \begin{cases} 1.0, & when\ x\ is\ a\ safety\ event \\ 0.8, & when\ x\ is\ an\ efficiency\ event \\ 0.5, & when\ x\ is\ an\ infotainment\ event \end{cases} \quad (9)$$

4.3.5. Sender Percentage

An event could be observed from more than one vehicle, so each of them sends an event message about the same event. When an event message is received, the receiver vehicle waits for a fixed period of time to receive other messages of the same event from other vehicles. This period is experimentally defined long enough in order to ensure vehicles could spread the event message across the network before being invalidated and could get messages about the event from their neighbours as much as possible. After the waiting period, the vehicle calculates the ratio of vehicles that send the same event message to the number of maximum neighbours. The percentage of vehicles sending the same event X to vehicle A, PE_A^X is defined as in Eq. 10:

$$PE_A^X = EN_A^X \ / \ MN_A \quad (10)$$

Event messages are considered as messages of the same event if their positions are the same. On the other hand, due to attackers, different types of event messages regarding to the event at the same position can be received. In other words, malicious vehicles are also included in the calculation of PE_A^X. To distinguish different types of events occurring at the same position, vehicles also count the number of vehicles that send event messages with the same event type at the same position and calculate its ratio to all vehicles that send an event message at this position. The percentage of vehicles sending the same event type X, to the vehicle A, PT_A^X is defined as in Eq. 11:

$$PT_A^X = TN_A^X \ / \ EN_A^X \quad (11)$$

Therefore, the event X in Eq. 10 corresponds to all received event messages at the same position regardless of their event types and the event X in Eq. 11 corresponds to messages that have the same position and the same event type.

4.3.6. Prior Knowledge

Vehicles take into account previous communications with the source of the received message. They use the last updated vehicle trust value about the

source when there exists a direct communication. In the case that another vehicle forwards the source vehicle's message, the receiver uses the vehicle trust value sent by the forwarder vehicle about the source and a coefficient which is its own vehicle trust value about the forwarder. A default trust value is used when there is no prior communication between the receiver and the source or forwarder vehicle. The same calculation is also done with the data trust value of the event message sent by the forwarder vehicle.

4.3.7. Majority Opinion

Vehicles calculate some average values to have knowledge about the opinion of majority. The average weight of the vehicles sending the same event, similar to the sender percentage, is calculated by dividing the total weight of vehicles that send an event message at the same position by the count of them. The average weight of the events at the same location is calculated using the count of vehicles that send the same event message, hence an idea about the opinion of majority for the event type is obtained. The average weight of vehicles sending the same event X to the vehicle A, VW_A^X is defined as in Eq. 12 and the average weight of events at the same location X sent to vehicle A EW_A^X is defined as in Eq. 13:

$$VW_A^X = (\sum_{i=1}^{EN_A^X} W_V^i) \ / \ EN_A^X \tag{12}$$

$$EW_A^X = (\sum_{i=1}^{EN_A^X} W_E^i) \ / \ EN_A^X \tag{13}$$

Vehicles calculate the average trust value of the source vehicle weighted by the vehicle trust values sent from forwarders and their own trust values about the senders. If a vehicle directly receives a message from its source, the receiver vehicle takes into account its own trust value about the source vehicle. When a vehicle receives a message from an intermediate/forwarder node, the receiver vehicle calculates the weighted vehicle trust value using the trust value sent from the forwarder vehicle about the source vehicle and its own trust value about the forwarder vehicle. The average weighted data trust value of the event message is also calculated based on the data trust value of the event message sent by the forwarder and the trust value of the receiver vehicle about the forwarder. The average weighted trust value of source vehicle S by vehicle R TV_R^S is defined as in Eq. 14 and the average

weighted trust value of event X by vehicle R TE_R^X is defined as in Eq. 15 (i means each forwarder vehicle):

$$TV_R^S = (\sum_{i=1}^{EN_A^X} VT_R^i * VT_i^S)/EN_A^X \tag{14}$$

$$TE_R^X = (\sum_{i=1}^{EN_A^X} VT_R^i * ET_i^X)/EN_A^X \tag{15}$$

4.3.8. Malicious Percentage

Vehicles keep track of malicious percentages of other vehicles from their own point of view. Each vehicle classifies received event messages either benign or malicious by calculating the data trust value of event messages according to the trust formula given in Section 4.4. They calculate the percentage of event messages predicted as malicious in all event messages sent by the source vehicle. If the receiver vehicles classify the event messages correctly, this ratio can play a significant role in distinguishing subsequent event messages.

4.4. Trust Calculation

The trust formula based on trust evidence is generated by evolutionary computation. Vehicles use this formula to calculate the data trust value of event messages received from their neighbour vehicles to decide whether the event message is malicious or benign. The values of evidences used in the generated formula are computed every time an event message is received. To prevent unnecessary computing overhead, the calculation of the trust value is made only if the values of the proximity evidences are in the determined limits. In addition, beacon messages are stored in a sliding window and stale messages are discarded to keep the memory consumption low.

Please note that all trust evidences used in the trust formula and the formula itself are simple calculations. Vehicles only send the trust values of the sender vehicle and data itself in event messages, hence the communication cost is negligible compared to event messages. On the contrary, a successful trust management system has a positive effect on the communication cost by eliminating untrusted messages from the network traffic. Other necessary information such as trust value of vehicles and a list of neighbor vehicles is stored in vehicles.

4.5. Trust Distribution

The dynamically changing topology of VANETs could cause vehicles to encounter with vehicles that they have not communicated before and had no experience about. Therefore, they should prefer to take into consideration the recommendations from their own trustee rather than deciding randomly to trust such newly encountered vehicles or not. Trust distribution plays a vital role to achieve that.

Vehicles only forward event messages that they have decided to be trustworthy. Before they forward the event messages, they add their opinions about them and their sender vehicle. This opinion contains both the data trust value of the event message and the vehicle trust value of its sender. Besides these two trust values, the following information about the forwarder vehicle are also added to the event message: its identifier, type, position, and velocity. Table 6 shows the forwarded event message format.

Table 6: Format of the Forwarded Event Message

Unique Identifier	Vehicle Type	Message Time	Current Position	Current Velocity
Event Type	Event Description	Event Position	Forwarder Identifier	Forwarder Type
Forwarder Position	Forwarder Velocity	Forwarder Data Trust	Forwarder Vehicle Trust	

Vehicles do not forward event messages that they do not trust. However, they inform other vehicles by sending their negative opinions about untrusted event messages and about the source vehicles that initiate such event messages. Hence, by distributing such information about attacks, they help to prevent further attacks from those source vehicles. The negative opinion message contains the identifier of the attacker vehicle, the data trust value of the malicious event message, and the vehicle trust value of the attacker in addition to information about the owner of this negative opinion. Table 7 shows the format of the negative opinion message.

If the receiver nodes of event messages correctly calculate the data trust value of these messages and classify the attacks correctly, they increase the detection possibility of these attacks by their neighbours even if they did

Table 7: Format of the Negative Opinion Message

Unique Identifier of Attacker	Vehicle Identifier	Vehicle Type	Current Position	Current Velocity
Vehicle Trust Value of Attacker		Data Trust Value of Malicious Message		

not meet the attacker before. Misclassification causes benign vehicles are regarded as attackers, and thus the fitness value decreases.

4.6. Trust Update

Vehicles keep the trust value of each vehicle they encounter in order to preserve the results of interactions with the sender vehicles and update these trust values after trust calculation of each event initiated by these source vehicles.

Event messages sent from vehicles that have higher trust value are decided more likely to be trustworthy than event messages from untrusted vehicles. The trust values of vehicles are initialized to a default value and updated according to the Eq. 16 every time an event message is received from these sender vehicles. Let us assume that the vehicle R receives an event message sent from vehicle S. Here, ET_R^S represents the trust value of the event message and TT refers to the threshold for accepting this event message to be forwarded. Where CT_R^S shows the current trust value of vehicle S calculated by the vehicle R, T_R^S indicates the newly updated vehicle trust value of vehicle S calculated by the vehicle R.

$$T_R^S = \begin{cases} CT_R^S \times ET_R^S, & 0 \leq ET_R^S < TT \\ CT_R^S + (1 - CT_R^S) \times (\dfrac{ET_R^S - TT}{1 - TT}), & TT \leq ET_R^S \leq 1 \end{cases} \tag{16}$$

While calculating T_R^S, a well-known principle about trust "hard to earn but easy to lose" [6, 7, 8] is applied. Increasing rate of a vehicle's trust value is proportional to the gap between the maximum trust value and the vehicle trust value, and the normalized trust value of the event message. In contrast, untrusted event messages will rapidly decrease the trust values of the source vehicles that send these messages.

Trust values of source vehicles are calculated in addition to the trust value calculation of forwarder vehicles. A node who receives an event message

updates the trust value of the source vehicle of this event message according to the Eq. 16 in addition to its sender. They pay attention to the opinions of forwarder vehicles while they update the vehicle trust values of source vehicles.

When vehicles receive a negative opinion, they update the trust value of the source vehicle using Eq. 17. A receiver vehicle R updates the trust value of the source vehicle S (T_R^S) by decreasing its current trust value (CT_R^S) using the data trust value of the event message sent by the source to the forwarder (ET_F^S) and the trust value of the source vehicle (CT_R^S) with a factor of the trust value of the forwarder vehicle F (CT_R^F) that sends the negative opinion.

$$T_R^S = CT_R^S - CT_R^F \times (CT_R^S \times (1 - ET_F^S)) \tag{17}$$

4.7. Evolutionary Dynamic Optimization

Genetic programming [34, 35] is a population-based search algorithm inspired by natural evolution. It starts with generating a population of individuals (usually at random) which are candidate solutions for the target problem. Then, each individual is evaluated using a fitness function and is assigned with a fitness value that indicates how well this candidate solves or comes close to solving the problem at hand. Until a termination criterion is satisfied, new populations are generated iteratively by using selection, crossover, and mutation operators, as in natural evolution. These genetic operators are used to provide better solutions in the new population. The pseudocode of a generic single-objective genetic programming is given Alg. 1.

Algorithm 1 The pseudocode of a generic single-objective GP

```
generate the initial population of individuals randomly
while a termination criterion is not satisfied do
    evaluate the fitness value of each individual
    select the fittest individuals for reproduction
    breed new individuals through genetic operators
    replace the least-fit individuals with new individuals
end while
return best-of-run individual
```

Here, each individual shows a candidate formula to be used for trust calculation and is represented as a tree in GP. Since the tree structure of GP is very suitable to represent the problem at hand, GP is preferred over

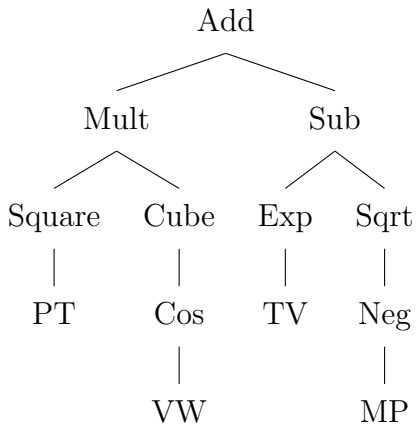

Figure 1: The GP tree of a simple trust formula including trust evidences and operations

other evolutionary computation algorithms in this study. In-order traversal of the tree outputs a candidate formula. Terminal nodes of the tree are trust evidences in Table 4 and some ephemeral random constants (ERC). Non-terminal nodes of the tree consist of the mathematical operations listed in Table 8. These operations are implemented to have the result value of $[0, 1]$. An example GP tree which represents a simple trust formula that uses some trust evidences and mathematical operations of the model is shown in Figure 1. This tree corresponds to the following formula given in Eq. 18.

$$\{PT^2 \times [(\cos(\pi \times VW)+1)/2]^3 + [(e^{TV}-1)/(e-1) - \sqrt{1-MP}+1]/2\}/2 \quad (18)$$

The initial population is generated randomly. A fitness value is assigned to each individual based on its detection rate of false and fake event messages. Higher value of fitness value shows better individuals, so the algorithm tries to increase the fitness value of the population using genetic operators. Selection operator probabilistically determines the parent individuals that will be used in the crossover and mutation operators. Better individuals have a higher chance to be selected. Crossover and mutation operators are used on the selected parents to breed new individuals. The crossover operator exchanges different portions of the parents and produces two new child individuals. It aims to create better solutions using good parts of parents. In the mutation operator, some portions of newly generated solutions are changed randomly to increase diversity and produce better solutions. GP terminates when the

Table 8: Genetic Programming Operation Set

Name	**Operation**
Add	$(X + Y) / 2$
Mult	$X \times Y$
Square	$X \times X$
Cube	$X \times X \times X$
Neg	$1 - X$
Sub	$(X - Y + 1) / 2$
Exp	$(e^X - 1) / (e - 1)$
Sqrt	$\sqrt{X}$
Sin	$(\sin(\pi X - (\pi / 2)) + 1) / 2$
Cos	$(\cos(\pi X) + 1) / 2$

ideal solution is found and returns it. Generally, finding the ideal solution takes a very long time for such complex problems. Thus, a predefined number of generations is used, so the GP terminates when it reaches that number of generations and returns the current best solution.

A vehicle makes a true positive (TP) decision if it correctly identifies a malicious event message as untrustworthy. Similarly, if a vehicle identifies a benign event message as trustworthy, it makes a true negative (TN) decision. A vehicle makes a false positive (FP) decision if it tags a benign event message as untrustworthy. Similarly, a malicious event message tagged as trustworthy is a false negative (FN) decision. Based on TP, TN, FP and FN values, the fitness value of the generated trust formula is calculated using Matthews Correlation Coefficient (MCC) [36], defined as in Eq. 19.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{19}$$

MCC is a widely used measure of the quality of binary classification in the machine learning field. It takes into account the four values in the confusion matrix equally weighted, thus it performs better when the positive and negative classes are imbalanced [37, 38, 39, 40]. It takes values in the interval $[-1, 1]$. The value 1 shows the perfect positive relationship and the

value -1 shows the perfect negative relationship. The value 0 represents no correlation, i.e., random prediction.

Evolutionary dynamic optimization aims to solve dynamic optimization problems by applying EC techniques. DOPs are defined as "are solved online by an optimization algorithm as time goes by" in [2]. They state that the fitness landscape of the dynamic problem changes in DOPs and the optimization algorithm must provide new optimal solutions. A dynamic trust management problem for VANETs is a good example of DOP. Vehicle and event densities are some examples that are subject to change over time of day. These affect directly to the fitness landscape of the dynamic trust management problem, so the trust calculation formula used as the solution must be changed to find a new optimal solution which classifies the event messages better.

The proposed dynamic trust management model tracks the fitness landscape of the DOP to detect a change in the VANET environment. The most common change detection approach in the literature is applied in this study by reevaluating the current best solution as the detector in the next generation [41, 42, 43]. The change in the fitness value of the detector means the change of the problem. When the dynamic fitness landscape of the problem is changed to an area in which the algorithm does not have members in the area that includes the new global optimum, the algorithm fails to track the moving global optimum and turns into tracking a local optimum because it is already converged and could not react to the change [2]. Crossover operator does not help the converged algorithm because this searches only around the local optimum, which makes it a kind of local search. Small changes of the fitness landscape are tracked by the mutation operator, and large changes are tracked by another operator which generates new random individuals in all search areas with the aim of finding a better solution than the current best, which is likely close to the moving global optimum [2]. It is shown in [44] that these adaptive control parameters help GP to perform better than the static control parameters in dynamic problems. When a change is detected, the model applies the commonly used EDO approach, which is diversity introducing, by simply increasing the diversity of GP by increasing the mutation rate and introducing random individuals to the population. This provides the algorithm can track the moving global optimum even if the dynamic fitness landscape moves to an area that the population has no individuals in it. This is also shown by the results in this study.

5. Experimental Settings

The proposed method is evaluated on several experimental scenarios in order to show its performance on varying conditions. Each experiment has two phases: evolving a trust formula on a network topology as the training phase and evaluating the trust formula on other similar network topologies as the testing phase. They are given in detail in this section.

5.1. Training Phase

In training, each evolved trust formula by GP is executed on the same set of networks in order to evaluate its fitness value. These formulas are used to classify application messages that vehicles get from their neighbours as either benign or malicious. A fitness value is assigned to them based on their classification performance using MCC. GP operators are then applied to evolve new formulas for trust calculation with the aim of generating fitter individuals at each generation. Networks used in training and testing are simulated by using the ns-3 network simulator [45]. The ECJ toolkit [46] is used for EC implementation.

5.1.1. Network Properties

The parameters used in the network simulations are listed in Table 9. The values of parameters other than the application-specific ones are chosen in accordance with a previous trust management model for VANETs [47]. Each candidate trust formula in GP is evaluated on three networks. In each network simulation, the random waypoint mobility model is applied to generate different network topologies and mobility patterns. In each network, the initial placements of vehicles and event messages are assigned randomly. Because of this randomness, each candidate solution can be evaluated on different networks with varying traffic and mobility patterns. The same three networks are used to evaluate the performance of each evolved trust formula, hence the fitness values of individuals are comparable. The fitness value of an individual, i.e., the trust formula, is calculated as the average of MCC values on these three networks. Each network simulation time is limited to 300 seconds to complete the whole evolution process in a reasonable time.

Two different sets of parameters are used for each scenario to introduce a change to the problem besides the dynamic nature of VANET topology. This change makes the problem a dynamic optimization problem by introducing more dynamicity over time. Each scenario has only one change point of the

Table 9: Network Simulation Parameters

Name	**Value**
Simulation area	600 m x 600 m
Number of vehicles	50, 100
Ratio of vehicles	5% with high trust, 15% with medium trust 80% with low trust
Ratio of attackers	10%
Training simulation time	300 seconds
Test simulation time	900 seconds
Vehicle placement	Random
Mobility model	Random waypoint
Vehicle speed	20 m/s
Number of events	25, 50, 100
Ratio of events	10% safety, 40% efficiency 50% infotainment
Event placement	Random
Event detection range	10 meters
Max event distance	50 meters
Max event time	1 second
Max delay time	0.2 seconds
Default trust value	0.5
Change in the problem	number of events (from low to high)

problem in time, which is introduced as an increase in the number of events. This causes two trust formulas are evolved in each scenario; one is just before the problem change, the other is at the end of the scenario after the problem change. Both of these solutions are evaluated in test environments and are

compared.

Each scenario has the same number of vehicles and attackers, the same ratio of events regardless of the problem change carried out in the middle of the simulation. Vehicles with high and medium trust make up, respectively 5% and 15% of the total vehicles and the rest of them are low level ordinary vehicles. Attacker vehicles are always chosen among the ordinary vehicles and their ratio is fixed along the scenario. Similarly, safety events make up 10% and efficiency events make up 40% of the total events, and the other events are infotainment events. The problem is changed by increasing the event number from 50 to 100 while preserving the ratios of all event types.

The running time of the evolving trust formula is proportional to the running time of GP, hence the size of individuals evaluated in each generation (I), the number of generations (G), and the cost of fitness evaluation of each trust formula (F) is used to determine the time complexity of the proposed approach which is defined as in Eq. 20. The average of running the trust formula on three network simulations is calculated for the cost of fitness evaluation of each trust formula. The time complexity of in-order traversal of the trust formula tree, whose number of nodes is n, is $O(n)$. Please note that the number of nodes in the tree is limited by the maximum tree depth as given in Table 10.

$$O(I \times G \times F) \tag{20}$$

5.1.2. EDO Properties

Table 10 lists the parameters used in the application of EDO technique. Each individual in the population represents a mathematical formula to calculate the trustworthiness value of application messages. An initial population of 100 individuals is generated randomly. The crossover and mutation operators of GP are applied to the population after all individuals are run on the network simulations and their fitness values are acquired.

The best individual of the population is transferred as the elite individual to the next generation to detect whether the problem is changed or not. If the problem is not changed, the fitness value of the elite individual remains the same as before. The change of the fitness value of the elite means that there occurs a problem change. This change detection mechanism is called "detecting change by reevaluating solutions" [2] and the use of the current best solution as the detector is a common approach [41, 42, 43].

The problem is changed at 50^{th} generation in the evolution process. If the solution of the new problem moves to an area that the population has

Table 10: EDO Parameters

Name	**Value**
Population size	100 individuals
Crossover probability	0.9 / 0.6
Mutation probability	0.1 / 0.3
Diversity probability	0.0 / 0.1
Elitism	The best individual of the population
Terminal nodes	Trust evidences and ERC
Non-terminal nodes	add, sub, mult, sin, cos, exp, square, sqrt, cube, neg
Generation size	50 + 50 generations
Maximum depth of tree	17

no individual in it, the algorithm needs to diverge to that area to track the moving optimum. Hence, the probabilities of crossover and mutation operators are changed to 0.6 and 0.3, respectively, in order to "introduce diversity when changes occur" [2]. A new diversity operator is introduced when the problem change is detected by EDO in order to increase the diversity by introducing new random individuals to the population besides the mutation. The evolution continues for another 50 generations to search solutions for the new problem based on the population of the prior problem rather than starting from scratch. Hence, the knowledge obtained in the first 50 generations are transferred to the new problem for evolving fitter solutions for the new problem. With this approach, it is expected to evolve better individuals in a shorter time than the traditional approach. Moreover, it is expected to produce higher initial and final performances in the new problem compared to learning from scratch.

5.2. Testing Phase

Each training phase produces two different formulas for the calculation of trust values of messages sent by vehicles: GP-based and EDO-based formula. These are the best known solutions to each problem in the experiment, the former is for the problem before change, and the latter is for the problem after change. EDO-based formula is tested on 100 simulated networks that

have the same number of vehicles and events, and the same event ratios, but with different network topologies and mobility patterns. The average of MCC values obtained from 100 runs is taken as the test result of the trust formula.

6. Experimental Results

The experimental results are presented in this section. The interpretations of the MCC values are selected from the three most commonly used ones based on different research areas, given at [48]. By taking into consideration that road safety and traffic efficiency are critical tasks, the most strict interpretations used in the medicine area are applied here, as shown in Table 11.

Table 11: Interpretation of the MCC Values

Perfect	1.0
Very Strong	0.8 - 1.0
Moderate	0.6 - 0.8
Fair	0.3 - 0.6
Poor	0.1 - 0.3
None	0.0 - 0.1

6.1. Performance of the Best Individuals

Figure 2 shows the fitness values of four individuals obtained in 20 different runs. The first one is the best individual obtained by GP only at the 50^{th} generation, just before the environment is changed. The second one is the same individual but evaluated on the new problem at 51^{th} generation just after the change is performed. These individuals are obtained by running the traditional GP algorithm only. The third and fourth ones are the best individuals obtained at the end of each run (i.e., at the 100^{th} generation) by EDO and GP, respectively. Please note that all individuals evolved are evaluated by using all three networks described in Section 5.1.1, so the fitness values in the Figure 2 are the average value of the results in the three networks.

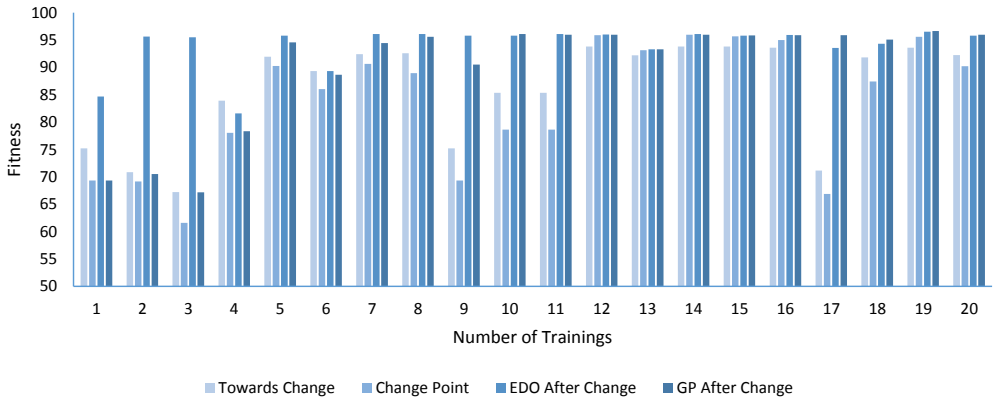

Figure 2: Performance of the best individuals before and after the change in the problem

As it is expected and shown in the figure, the fitness value of the best individual at 50^{th} generation decreases when it is applied to the new problem. Thus, searching a new individual becomes a necessity when the problem is changed. Only a couple of individuals increase their fitness value when the problem changes, but they already have high fitness values at this point. It can be concluded that the effect of the problem change is minimal when the best individual has already high fitness value and has a better convergence. When the final performances of GP and EDO are compared, it is shown that EDO mostly finds either better or equal individuals compared to GP. Besides, the EDO finds fitter individuals quicker than the GP as shown in the convergence graphs in Figure 3, which shows the fitness value of the best individual in each generation after the problem change point.

6.2. Performance on Networks with Higher Density of Benign Vehicles

Figure 4 shows the evaluation of the best individual on testing networks in which the total number of vehicles is increasing but the number of malicious vehicles is fixed. The increase in the density of vehicles can be corresponded to networks at different times. The density of vehicles increases at rush hours in urban areas and decreases after a while in the real world. Malicious messages (%) show the actual percentage of malicious messages in all messages in the network.

As shown in the figure, the average of MCC values starts from a very strong correlation level and goes down to a moderate correlation level towards 200 vehicles (quadruple of the initial density) and a fair correlation

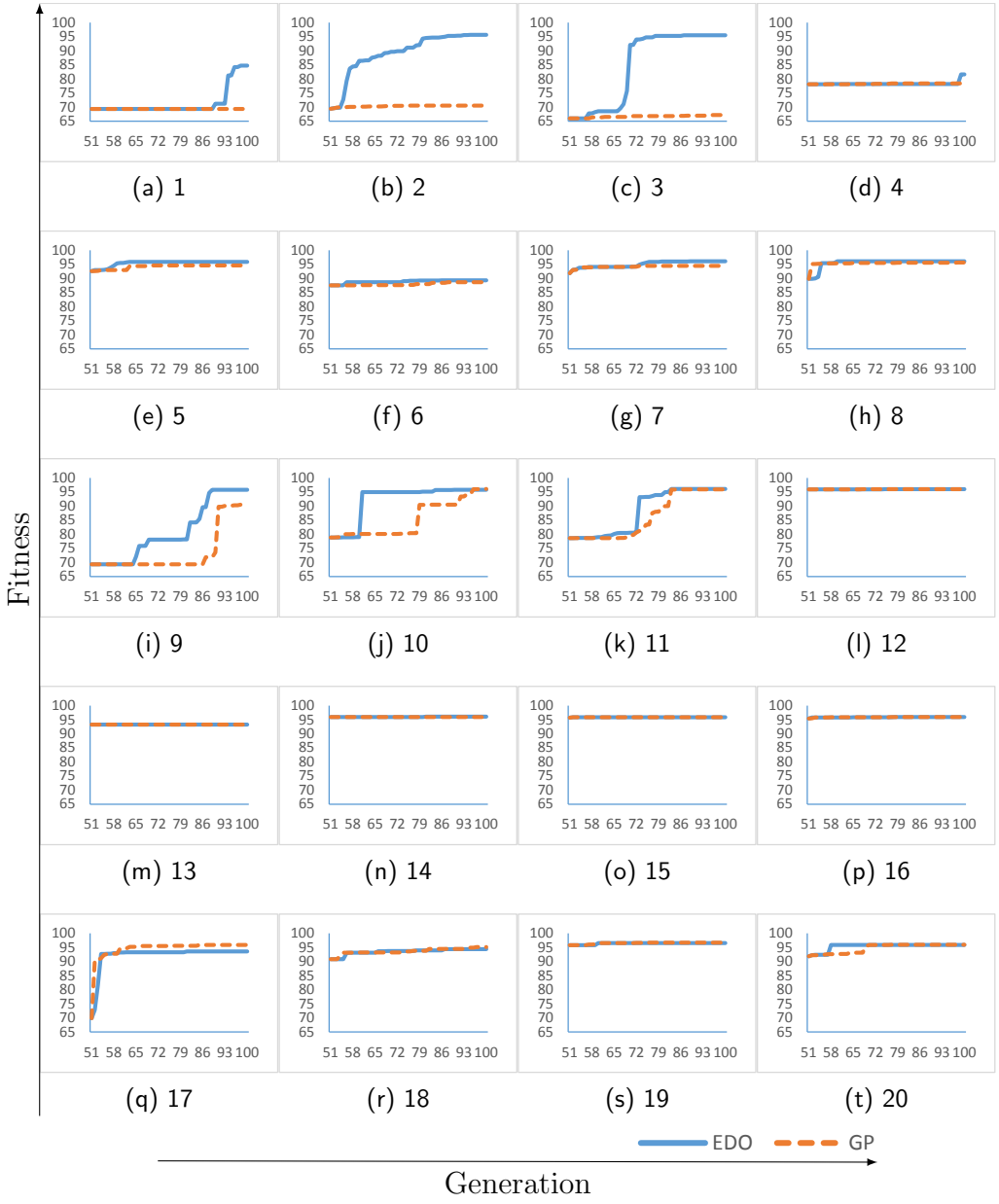

Figure 3: Convergence graphs of all training phases after 50^{th} generation

level afterwards even if the detection rate (DR) (i.e., recall) decreases only

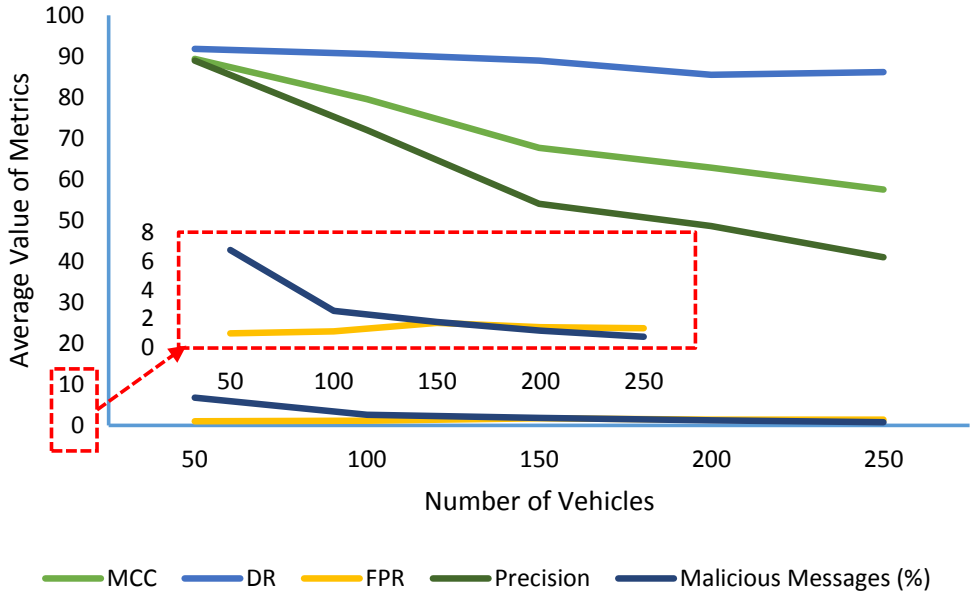

Figure 4: Performance of the model on networks with higher density of benign vehicles

about 5% while the vehicle number is increasing. Moreover, the average of false positive rates (FPR) only fluctuates between 1.01% and 1.73%, while the percentage of malicious messages in the total messages decreases from 6.76% to 0.77%, but the mean percentage of precision value decreases. Because while the number of vehicles increases, the total number of benign event messages in the environment increases dramatically and the formula produces more FPs for the sake of detection of malicious messages. In such rush hours, the event message produced by one vehicle is delivered to more vehicles. With the help of forwarding benign messages, messages classified as TN gradually increase. The formula produces more FPs in such a case. Because the increase rates of both FPs and TNs are similar, the FPR does not change so much. In contrast, TPs increase much slower than FPs, thus the precision decreases. Each FN message is forwarded to other vehicles and unless all vehicles detect the malicious message correctly, it continues to be forwarded in the environment. On the other hand, each FP message is dropped immediately to prevent the propagation of the message that is reputed to be malicious. Thus, the model is evolved towards to accept misclassifying some benign messages in order not to miss any attack.

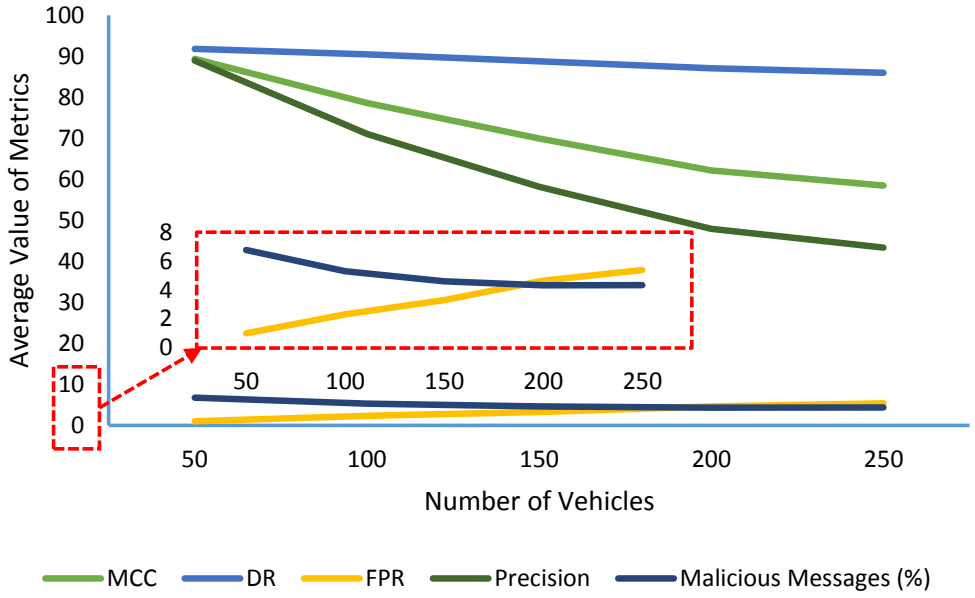

Figure 5: Performance of the model on networks with higher density of vehicles & attackers

6.3. Performance on Networks with Higher Density of Vehicles & Attackers

Figure 5 shows the evaluation of the best individual on testing networks, where the number of total vehicles and malicious vehicles are increasing, preserving the initial ratio of attackers on all networks. Similar to Figure 4, the average of MCC values starts from a very strong correlation level and goes down to a moderate correlation level towards 200 vehicles and a fair correlation level afterwards even if the DR decreases only about 5% while the vehicle number is increasing. Although the attacker ratio does not change, the percentage of malicious messages in the total messages decreases from 6.76% to 4.33% like in the Figure 4 but not that much because of the increase in the number of malicious vehicles in this scenario. Moreover, the average percentage of FPR increases from 1.01% to 5.38%, and the average percentage of precision value decreases. While benign vehicles begin to detect malicious messages and isolate the attackers throughout the scenario, preserving the attacker ratio and increasing the malicious vehicle proportional to it does not cause to increase the malicious message ratio. Hereby, the results of this scenario are similar to the previous one. As it is stated above, the model is inclined to produce formulas where they output more FPs for the sake of

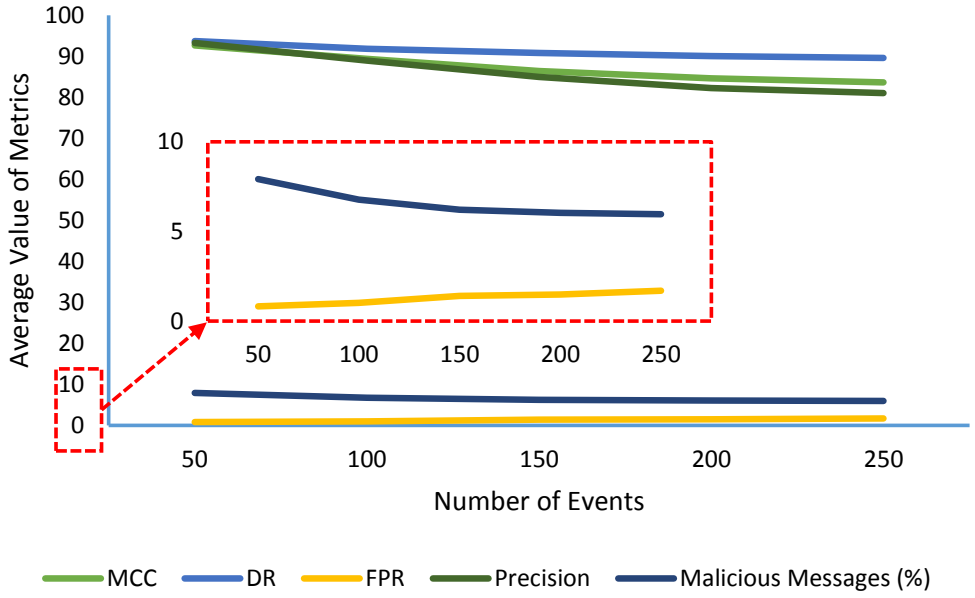

Figure 6: Performance of the model on networks with higher density of events

Table 12: Values of the Metrics in Figure 6

Event Number	MCC	DR	FPR	Precision	Malicious Message
50	92.65%	93.72%	0.82%	93.26%	7.92%
100	89.38%	91.81%	1.01%	88.95%	6.76%
150	86.45%	90.83%	1.39%	84.93%	6.21%
200	84.63%	90.03%	1.48%	82.22%	6.03%
250	83.64%	89.60%	1.69%	81.03%	5.96%

detection of malicious messages while the total benign event messages in the environment increases.

6.4. Performance on Networks with Higher Density of Events

Figure 6 shows the performance of the best individual on simulated networks with higher density of event messages. The average of MCC values again starts from a very strong correlation level and maintains its correlation

level even if its own value and the precision rate slowly decrease while the number of events is increasing (Table 12). The average DR decreases about 4% and FPR only increases from 0.82% to 1.69%. The percentage of malicious messages in the total messages only decreases from 7.92% to 5.96% unlike previous scenarios, because the increase in the number of events causes an increase in the number of both benign and malicious messages. This results in a slight decrease in the fitness value compared to Figures 4 and 5 as the model does not need to misclassify many benign messages in order to detect malicious messages.

The high density of event messages simulates unusual conditions on the road that are not seen everyday, such as road maintenance and closed roads in an area. In such cases, vehicles send/forward more event messages than before. Because malicious vehicles modify event messages about real events and forward these modified malicious messages to the network, the ratio of malicious messages in the network does not decrease much. This gives the model to detect malicious messages without increasing its error comparing to the previous scenarios where the density of vehicles/attackers is increased. As we compare Figure 6 with Figures 4 and 5, it can be said that the model separates benign and malicious messages better when there is an adequate amount of malicious messages rather than low or limited attacks while maintaining a successful attack detection mechanism on any environment.

6.5. Performance on Networks with Higher Density of Attackers

Figure 7 shows the performance of the best individual on networks having more number of malicious vehicles. The increase in the number of attackers causes more false information to be distributed in the network. Differently from the other testing scenarios, the percentage of malicious messages in the network increases proportionally to the increase in the number of malicious vehicles in this scenario. The evolved model generally does not miss malicious messages but misclassifies some benign messages as shown in the previous test scenarios. Therefore, increase in the number of malicious messages does not affect the detection, they could be easily detected by the model. Hence, the MCC mean value again starts from a very strong correlation level and maintains its correlation level unlike other test scenarios that are increasing the density of vehicles/attackers. Additionally, different from all other test scenarios, the precision rate increases slowly and even the DR increases slightly while the malicious vehicle number is increasing. Moreover, the mean

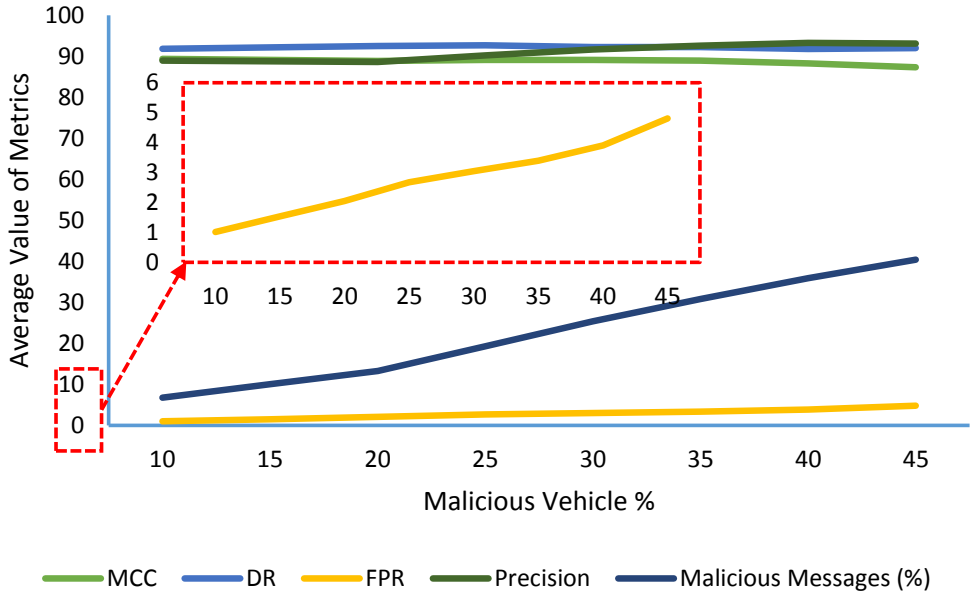

Figure 7: Performance of the model on networks having more malicious nodes

percentage of FPR only increases from 1.01% to 4.80% because of the decreasing number of benign messages. As shown in the Figure 6 that the fitness value is maintained at a level when the ratio of malicious messages does not decrease much, Figure 7 shows also that increasing the malicious message ratio can help the model to maintain the fitness level. As a result, increase in the number of malicious vehicles does not decrease the fitness value as the model already tends to detect malicious messages, thus this results in more TPs and less FPs.

These results are compared with ART [47], which is implemented in a similar attack scenario as stated in Section 5.1.1. When more attacks exist in the network, the precision and recall values of ART decrease as stated in [47]. The precision and recall are decreased from ≈93%, ≈91% to ≈87%, ≈85.1%, respectively in ART, when the number of malicious vehicles increased from 15% to 40%. Therefore, the proposed approach shows much better performance than ART in a network, where more than 25% activities are malicious as shown in Table 13. To sum up, the experiments show that the model is very robust to the increase in malicious vehicles. This is an essential characteristic for a trust management model in VANETs, since

Table 13: Values of the Metrics in Figure 7

Malicious Vehicle	MCC	DR / Recall	FPR	Precision	Malicious Message
9.42%	89.38%	91.81%	1.01%	88.95%	6.76%
17.42%	88.83%	92.54%	2.05%	88.62%	13.26%
24.52%	89.13%	92.68%	2.67%	90.18%	19.30%
31.22%	89.09%	92.25%	3.04%	91.66%	25.37%
36.84%	88.94%	92.26%	3.39%	92.59%	30.85%
41.80%	88.25%	91.79%	3.90%	93.27%	35.87%
46.32%	87.33%	92.01%	4.80%	93.06%	40.41%

Table 14: Real World Application Simulation Parameters

Name	**Value**
Simulation area	4.6 km x 3.0 km street map
Number of vehicles	99 (low), 210 (medium), 370 (high)
Vehicle mobility/speed	real-world traffic data model
Number of events	100
Event detection range	100 meters
Max event distance	500 meters
Change in the problem	number of vehicles (from medium to high)

misclassifying of a malicious attacker could result in drastic results in traffic.

7. Real World Application Case Study

To reduce the gap between synthetic environments with real-world applications, the proposed dynamic trust management model is also run on a real-world traffic model taken from a street map in Zurich. In this simulation of a real-world application, the initial position, mobility, and speed of vehicles are simulated according to the real-world traffic model [49] which is included in the distribution of ns-3. It has three options of traffic density

settings as low, medium, and high and takes 300 seconds. The parameters of this simulation that are different from the Table 9 are listed in Table 14.

The fitness value of the best individuals in the training of this experiment in the format of Figure 2 is as follows: 77.80% at towards change, 70.30% at change point, 96.11% at EDO after change, and 93.24% at GP after change. As shown from the results, the model has similar outcomes to the previous experimental scenarios on a real-world traffic model.

The best individual found by EDO is tested in 300 different environments that have the same street map and real-world traffic data model but different event positions which are placed randomly. The average values of the five metrics of these test results that are given in the previous test results are as follows: 80.68% MCC, 72.27% DR, 0.14% FPR, 91.57% precision and 2.04% malicious messages. As also shown from these test results, the model has again good outcomes on a real-world traffic model. It is stated that in a machine learning-based botnet detection study [50], machine learning algorithms that reach 99% detection rates on synthetic environments could have a DR value of 75% on real-world environments, which can explain the drop in the DR value of this study. The very strong correlation level of the MCC value shows clearly that the proposed dynamic trust management model is also effective against bogus information attacks on the real-world traffic model and could be used in real-world applications.

8. Limitations and Future Works

Traditionally, the solutions in the literature propose a predefined static trust calculation formula for VANETs. However, in this study, the evolution of a trust formula that adapts to changes in the environment is proposed to be able to change the trust calculation dynamically. Even though the proposed approach is suitable for dynamic environments such as VANETs, it requires to detect changes in the problem in order to adapt them. Here, the decrease in the fitness value (MCC) is used to do that. However, if the attackers know the fitness function, they might try to evade from it. Additionally, it is not trivial to decide the change rate of GP operators in case of a change in the environment, since changing parameters too little will result in local search, and changing too much will result in random search [2].

Although EC and EDO techniques are employed in this study, some other algorithms can be used to build a trust management model automatically in

VANETs. As support vector machines are reliable machine learning techniques for non-linear classification scenarios, trust management models can be developed using them [14]. Additionally, it is shown that reinforcement learning (RL) is a promising approach for processing large amounts of data sent from vehicles in VANETs [16]. In the future, the use of deep reinforcement learning techniques on the problem could be explored. Chen et al. [51] propose two new strategies in order to stabilize the value estimation, hence to mitigate the unstable reward estimation problem of Deep RL in dynamic environments. Furthermore, transfer learning could be investigated in order to adapt the model to a new, more dynamic environment in the future.

This study uses two exemplar attacks in order to show the performance of the proposed trust management model. In the future, more complex attack scenarios such as on-and-off and collaborative attacks can be implemented. During an on-and-off attack scenario, malicious vehicles cease executing their attacks for a short time and become trusted by other vehicles in the network by behaving benignly in that period. Malicious vehicles might support one another in collaborative attacks by sending malicious messages of attackers to other vehicles with high data trust values. The detection of malicious vehicles in such cases becomes more challenging, which needs further investigation.

9. Conclusion

This paper presents the first research that explores the use of evolutionary computation techniques and evolutionary dynamic optimization algorithms to the dynamic trust management problem in VANETs. A dynamic trust management model based on genetic programming and EDO is proposed to evaluate the trustworthiness of messages about events on the road sent by vehicles in VANETs automatically. The trustworthiness of vehicles are tracked using the vehicle trust value based on the data trust values of their event messages to establish a more reliable trust management framework with the combined trust model. A large number of trust evidences are collected from messages in the network to represent the complex properties of VANETs, including the dynamicity. This set covers much more trust evidence than other trust management studies in the literature. A trust formula based on the trust evidence set is evolved by genetic programming and later is adapted to the dynamically changing network conditions by EDO. The simulation results show that the proposed dynamic trust management model is effective against bogus information attacks.

References

[1] F. Sakiz, S. Sen, A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV, Ad Hoc Networks 61 (2017) 33–50. doi:10.1016/j.adhoc.2017.03.006.

[2] T. T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, Swarm and Evolutionary Computation 6 (2012) 1–24. doi:10.1016/j.swevo.2012.05.001.

[3] B. Dorronsoro, P. Ruiz, G. Danoy, Y. Pigné, P. Bouvry, Evolutionary algorithms for mobile ad hoc networks, John Wiley & Sons, 2014. doi:10.1002/9781118833209.

[4] J.-H. Cho, A. Swami, R. Chen, A survey on trust management for mobile ad hoc networks, IEEE Communications Surveys & Tutorials 13 (4) (2011) 562–583. doi:10.1109/SURV.2011.092110.00088.

[5] K. Govindan, P. Mohapatra, Trust computations and trust dynamics in mobile adhoc networks: A survey, IEEE Communications Surveys & Tutorials 14 (2) (2012) 279–298. doi:10.1109/SURV.2011.042711.00083.

[6] S. Ma, O. Wolfson, J. Lin, A survey on trust management for intelligent transportation system, in: Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science, ACM, 2011, pp. 18–23. doi:10.1145/2068984.2068988.

[7] H. Yu, Z. Shen, C. Miao, C. Leung, D. Niyato, A survey of trust and reputation management systems in wireless communications, Proceedings of the IEEE 98 (10) (2010) 1755–1772. doi:10.1109/JPROC.2010.2059690.

[8] J. Zhang, A survey on trust management for VANETs, in: 2011 IEEE International Conference on Advanced Information Networking and Applications, IEEE, 2011, pp. 105–112. doi:10.1109/AINA.2011.86.

[9] Y.-M. Chen, Y.-C. Wei, A beacon-based trust management system for enhancing user centric location privacy in VANETs, Journal of Communications and Networks 15 (2) (2013) 153–163. doi:10.1109/JCN.2013.000028.

[10] C. J. Van Rijsbergen, Information Retrieval, Butterworths, London, UK, 1979, Last accessed 31 May 2022.
URL http://www.dcs.gla.ac.uk/Keith/Preface.html

[11] X. Yao, X. Zhang, H. Ning, P. Li, Using trust model to ensure reliable data acquisition in VANETs, Ad Hoc Networks 55 (2017) 107–118. doi:10.1016/j.adhoc.2016.10.011.

[12] R. Hussain, J. Lee, S. Zeadally, Trust in VANET: A survey of current solutions and future research opportunities, IEEE transactions on intelligent transportation systems 22 (5) (2020) 2553–2571. doi:10.1109/TITS.2020.2973715.

[13] A. Mchergui, T. Moulahi, S. Zeadally, Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs), Vehicular Communications (2021) 100403doi:10.1016/j.vehcom.2021.100403.

[14] E. A. Shams, A. Rizaner, A. H. Ulusoy, Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks, Computers & Security 78 (2018) 245–254. doi:10.1016/j.cose.2018.06.008.

[15] N. Fan, C. Q. Wu, On trust models for communication security in vehicular ad-hoc networks, Ad Hoc Networks 90 (2019) 101740. doi:10.1016/j.adhoc.2018.08.010.

[16] D. Zhang, F. R. Yu, R. Yang, L. Zhu, Software-defined vehicular networks with trust management: A deep reinforcement learning approach, IEEE Transactions on Intelligent Transportation Systems (2020). doi:10.1109/TITS.2020.3025684.

[17] S. Hakak, T. R. Gadekallu, S. P. Ramu, P. K. R. Maddikunta, C. de Alwis, M. Liyanage, et al., Autonomous vehicles in 5g and beyond: A survey, arXiv preprint arXiv:2207.10510 (2022). doi:10.48550/arXiv.2207.10510.

[18] C. Ravi, A. Tigga, G. T. Reddy, S. Hakak, M. Alazab, Driver identification using optimized deep learning model in smart transportation, ACM Transactions on Internet Technology 22 (4) (2022) 1–17. doi:10.1145/3412353.

[19] M. Aslan, S. Sen, Evolving trust formula to evaluate data trustworthiness in VANETs using genetic programming, in: International Conference on the Applications of Evolutionary Computation (Part of EvoStar), Springer, 2019, pp. 413–429. doi:10.1007/978-3-030-16692-2_28.

[20] B. Dorronsoro, P. Ruiz, G. Danoy, Y. Pigné, P. Bouvry, Survey on Optimization Problems for Mobile Ad Hoc Networks, John Wiley & Sons, 2014, Ch. 3, pp. 49–78. doi:10.1002/9781118833209.ch3.

[21] D. G. Reina, P. Ruiz, R. Ciobanu, S. Toral, B. Dorronsoro, C. Dobre, A survey on the application of evolutionary algorithms for mobile multihop ad hoc network optimization problems, International Journal of Distributed Sensor Networks 12 (2) (2016) 2082496. doi:10.1155/2016/2082496.

[22] J. Kusyk, M. U. Uyar, C. S. Sahin, Survey on evolutionary computation methods for cybersecurity of mobile ad hoc networks, Evolutionary Intelligence 10 (3) (2018) 95–117. doi:10.1007/s12065-018-0154-4.

[23] S. Sen, A survey of intrusion detection systems using evolutionary computation, in: Bio-inspired computation in telecommunications, Elsevier, 2015, pp. 73–94. doi:10.1016/B978-0-12-801538-4.00004-5.

[24] S. Sen, J. A. Clark, Evolutionary computation techniques for intrusion detection in mobile ad hoc networks, Computer Networks 55 (15) (2011) 3441–3457. doi:10.1016/j.comnet.2011.07.001.

[25] U. E. Tahta, S. Sen, A. B. Can, GenTrust: A genetic trust management model for peer-to-peer systems, Applied Soft Computing 34 (2015) 693–704. doi:10.1016/j.asoc.2015.04.053.

[26] M. Sipper, R. S. Olson, J. H. Moore, Evolutionary computation: the next major transition of artificial intelligence?, BioData Mining 10 (1) (2017) 1–3. doi:10.1186/s13040-017-0147-3.

[27] H. Cheng, S. Yang, Genetic algorithms with immigrants schemes for dynamic multicast problems in mobile ad hoc networks, Engineering Applications of Artificial Intelligence 23 (5) (2010) 806–819. doi:10.1016/j.engappai.2010.01.021.

[28] H. Cheng, S. Yang, Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks, in: European conference on the applications of evolutionary computation, Springer, 2010, pp. 562–571. doi:10.1007/978-3-642-12239-2_58.

[29] H. Cheng, S. Yang, J. Cao, Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc networks, Expert Systems with Applications 40 (4) (2013) 1381–1392. doi:10.1016/j.eswa.2012.08.050.

[30] H. Cheng, S. Yang, Genetic algorithms for dynamic routing problems in mobile ad hoc networks, in: Evolutionary Computation for Dynamic Optimization Problems, Springer, 2013, pp. 343–375. doi:10.1007/978-3-642-38416-5_14.

[31] D. M. Chitty, M. L. Hernandez, A hybrid ant colony optimisation technique for dynamic vehicle routing, in: Genetic and Evolutionary Computation Conference, Springer, 2004, pp. 48–59. doi:10.1007/978-3-540-24854-5_5.

[32] L.-N. Xing, P. Rohlfshagen, Y.-W. Chen, X. Yao, A hybrid ant colony optimization algorithm for the extended capacitated arc routing problem, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41 (4) (2011) 1110–1123. doi:10.1109/TSMCB.2011.2107899.

[33] M. Mavrovouniotis, S. Yang, Ant colony optimization with immigrants schemes for the dynamic travelling salesman problem with traffic factors, Applied Soft Computing 13 (10) (2013) 4023–4037. doi:10.1016/j.asoc.2013.05.022.

[34] J. R. Koza, Genetic programming: on the programming of computers by means of natural selection, Vol. 1, The MIT Press, Cambridge, MA, 1992.

[35] J. R. Koza, Genetic programming as a means for programming computers by natural selection, Statistics and computing 4 (2) (1994) 87–112. doi:10.1007/BF00175355.

[36] B. W. Matthews, Comparison of the predicted and observed secondary structure of T4 phage lysozyme, Biochimica et Biophysica Acta (BBA)-Protein Structure 405 (2) (1975) 442–451. doi:10.1016/0005-2795(75)90109-9.

[37] S. Boughorbel, F. Jarray, M. El-Anbari, Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric, PloS one 12 (6) (2017) e0177678. doi:10.1371/journal.pone.0177678.

[38] D. Chicco, Ten quick tips for machine learning in computational biology, BioData mining 10 (1) (2017) 1–17. doi:10.1186/s13040-017-0155-3.

[39] D. Chicco, G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC genomics 21 (1) (2020) 1–13. doi:10.1186/s12864-019-6413-7.

[40] D. Chicco, N. Tötsch, G. Jurman, The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation, BioData mining 14 (1) (2021) 1–22. doi:10.1186/s13040-021-00244-z.

[41] X. Hu, R. C. Eberhart, Adaptive particle swarm optimization: detection and response to dynamic systems, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1666–1670. doi:10.1109/CEC.2002.1004492.

[42] X. Li, J. Branke, T. Blackwell, Particle swarm with speciation and adaptation in a dynamic environment, in: Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006, pp. 51–58. doi:10.1145/1143997.1144005.

[43] G. R. Kramer, J. C. Gallagher, Improvements to the *CGA enabling online intrinsic evolution in compact EH devices, in: NASA/DoD Conference on Evolvable Hardware, 2003. Proceedings., IEEE, 2003, pp. 225–231. doi:10.1109/EH.2003.1217670.

[44] M. Riekert, K. M. Malan, A. Engelbrect, Adaptive genetic programming for dynamic classification problems, in: 2009 IEEE congress on evolutionary computation, IEEE, 2009, pp. 674–681. doi:10.1109/CEC.2009.4983010.

[45] The NS-3 Consortium, NS-3 a discrete-event network simulator for internet systems, Last accessed 31 May 2022 (2008).
URL https://www.nsnam.org/

[46] S. Luke, ECJ A Java-based Evolutionary Computation Research System, Last accessed 31 May 2022 (1998).
URL https://cs.gmu.edu/~eclab/projects/ecj/

[47] W. Li, H. Song, ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks, IEEE transactions on intelligent transportation systems 17 (4) (2015) 960–969. doi:10.1109/TITS.2015.2494017.

[48] H. Akoglu, User's guide to correlation coefficients, Turkish journal of emergency medicine 18 (3) (2018) 91–93. doi:10.1016/j.tjem.2018.08.001.

[49] V. Naumov, R. Baumann, T. Gross, An evaluation of inter-vehicle ad hoc networks based on realistic vehicular traces, in: Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing, ACM, 2006, pp. 108–119. doi:10.1145/1132905.1132918.

[50] E. B. Beigi, H. H. Jazi, N. Stakhanova, A. A. Ghorbani, Towards effective feature selection in machine learning-based botnet detection approaches, in: 2014 IEEE Conference on Communications and Network Security, IEEE, 2014, pp. 247–255. doi:10.1109/CNS.2014.6997492.

[51] S.-Y. Chen, Y. Yu, Q. Da, J. Tan, H.-K. Huang, H.-H. Tang, Stabilizing reinforcement learning in dynamic environment with application to online recommendation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 1187–1196. doi:10.1145/3219819.3220122.