EVOLUTIONARY DYNAMIC OPTIMIZATION FOR DYNAMIC TRUST MANAGEMENT IN VEHICULAR AD HOC NETWORKS

ARAÇSAL TASARSIZ AĞLARDA DİNAMİK GÜVEN YÖNETİMİ İÇİN EVRİMSEL DİNAMİK ENİYİLEME

MEHMET ASLAN

PROF. DR. SEVIL ŞEN

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment of the Requirements

for the Award of the Degree of Doctor of Philosophy

in Computer Engineering

June 2023

ABSTRACT

EVOLUTIONARY DYNAMIC OPTIMIZATION FOR DYNAMIC TRUST MANAGEMENT IN VEHICULAR AD HOC NETWORKS

Mehmet ASLAN

Doctor of Philosophy, Computer Engineering

Supervisor: Prof. Dr. Sevil ŞEN

June 2023, 129 pages

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 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

ÖZET

ARAÇSAL TASARSIZ AĞLARDA DİNAMİK GÜVEN YÖNETİMİ İÇİN EVRİMSEL DİNAMİK ENİYİLEME

Mehmet ASLAN

Doktora, Bilgisayar Mühendisliği

Danışman: Prof. Dr. Sevil ŞEN

Haziran 2023, 129 sayfa

Araçsal tasarsız ağlarda (VANET'ler) güven yönetimi, merkezi olmayan, altyapısız ve dinamik olarak değişen topolojileri nedeniyle zorlu bir dinamik eniyileme problemidir. Evrimsel hesaplama (EC) algoritmaları, ortamdaki değişikliklerin bir sonucu olarak meydana gelen biyolojik evrimden ilham aldıkları için dinamik eniyileme problemlerini (DOP'lar) çözmek için iyi adaylardır. Bu çalışmada, VANET'ler için dinamik bir güven yönetimi modeli oluşturmak üzere genetik programlama (GP) algoritması ve evrimsel dinamik eniyileme (EDO) tekniklerinin kullanımını araştırıyoruz. Önerilen dinamik güven yönetimi modeli, sahte bilgi saldırılarını da içeren deneysel senaryoların simülasyonunda araçların ve mesajlarının güvenilirliğini uygun bir şekilde değerlendirir. Simülasyon sonuçları, evrimleşen güven hesaplama formülünün sahte mesajların VANET'ler üzerinde yayılmasını başarıyla engellediğini ve dinamik güven yönetimi modelinin problemdeki değişiklikleri tespit ederek zamanında tepki verdiğini göstermektedir. En iyi evrimleşen formül, ağ trafiğinin $\approx$ %5'i kötü amaçlı olduğunda, %89,38 Matthews Korelasyon Katsayısı (MCC), %91,81 tespit oranı (DR) ve %1,01 yanlış pozitif oranına (FPR) ulaşır. Formül, ağ trafiğinin $\approx$ %40'ı kötü amaçlı olduğunda %87,33 MCC, %92,01 DR ve %4,8 FPR elde ederek artan kötü amaçlı iletilere karşı dayanıklılığını gösterir. Önerilen model aynı zamanda gerçek

dünya trafik modeli üzerinde çalıştırılmakta ve yüksek MCC ve düşük FPR değerleri elde etmektedir. Bildiğimiz kadarıyla bu çalışma, EC ve EDO teknikleri kullanarak VANET'lerde dinamik güven yönetimi için otomatik olarak bir güven formülü oluşturan ilk uygulamadır.

Keywords: Araçsal Tasarsız Ağlar, Güvenlik, Güven Yönetimi, Evrimsel Hesaplama, Genetik Programlama, Evrimsel Dinamik Eniyileme

ACKNOWLEDGEMENTS

I would first like to express my gratitude to my thesis supervisor Prof. Dr. Sevil ŞEN for the continuous guidance and engagement through my PhD study and research. I have received a great deal of assistance throughout the writing of this thesis. Her expertise was very helpful in all stages of this work.

I would like to thank to the members of the thesis supervisory committee, Prof. Dr. M. Ali AKCAYOL and Prof. Dr. Ahmet Burak CAN, for their valuable feedback. I would also like to thank to the members of the thesis defense jury, Prof. Dr. Şebnem BAYDERE and Prof. Dr. Suat ÖZDEMİR, for their insightful comments and valuable questions.

I owe more than thanks to my family members: my parents Melahat and Salih, and my brother Süleyman for all their love, encouragement, and support throughout my life. They are always so helpful to me in numerous ways and encouraging me in whatever I pursue. Finally, huge thanks to Fatma who has always been there for me along the way. It would not have been possible for me to successfully complete this work without their guidance and support.

CONTENTS

Page

TABLES

FIGURES

ABBREVIATIONS

ABC	:	**A**rtificial **B**ee **C**olony
ACO	:	**A**nt **C**olony **O**ptimization
AI	:	**A**rtificial **I**ntelligence
AIS	:	**A**rtificial **I**mmune **S**ystems
ANN	:	**A**rtificial **N**eural **N**etwork
DL	:	**D**eep **L**earning
DOPs	:	**D**ynamic **O**ptimization **P**roblem**s**
DR	:	**D**etection **R**ate
DTNs	:	**D**elay **T**olerant **N**etwork**s**
EA	:	**E**volutionary **A**lgorithms
EC	:	**E**volutionary **C**omputation
ECJ	:	**E**volutionary **C**omputation Research System in **J**ava
EDO	:	**E**volutionary **D**ynamic **O**ptimization
EG	:	**E**volutionary **G**ames
EP	:	**E**volutionary **P**rogramming
ERC	:	**E**phemeral **R**andom **C**onstants
ES	:	**E**volution **S**trategy
FANETs	:	**F**lying **A**d hoc **NET**work**s**
FN	:	**F**alse **N**egative
FP	:	**F**alse **P**ositive
FPR	:	**F**alse **P**ositive **R**ate
GA	:	**G**enetic **A**lgorithm
GE	:	**G**rammatical **E**volution
GOA	:	**G**rasshopper **O**ptimization **A**lgorithm
GP	:	**G**enetic **P**rogramming
InVANETs	:	**In**telligent **V**ehicular **A**d hoc **NET**work**s**

IoV	:	**I**nternet **o**f **V**ehicles
ITS	:	**I**ntelligent **T**ransportation **S**ystems
MANETs	:	**M**obile **A**d hoc **NET**work**s**
MCC	:	**M**atthews **C**orrelation **C**oefficient
MGN	:	**M**obile **G**rid **N**etwork
ML	:	**M**achine **L**earning
NIOA	:	**N**ature-**I**nspired **O**ptimization **A**lgorithms
P2P	:	**P**eer-to-**P**eer
PSO	:	**P**article **S**warm **O**ptimization
RSU	:	**R**oad **S**ide **U**nit
SI	:	**S**warm **I**ntelligence
TN	:	**T**rue **N**egative
TP	:	**T**rue **P**ositive
V2I	:	**V**ehicle-to-**I**nfrastructure
V2V	:	**V**ehicle-to-**V**ehicle
V2X	:	**V**ehicle-to-Everything
VANETs	:	**V**ehicular **A**d hoc **NET**work**s**
WSN	:	**W**ireless **S**ensor **N**etwork

1. INTRODUCTION

Vehicles have been equipped with various smart modules to ensure safer, efficient, and reliable road transportation in recent years. These smart modules are forming intelligent transportation systems (ITS) that cover different aspects of transportation and traffic management. Some examples of ITS applications are navigation system, driving assistance, and parking guidance. The need of ITS technology increases rapidly proportional to the increasing use of road transportation. This leads to a growth of the research area about ITS technology and its applications.

1.1. Vehicular Ad Hoc Networks (VANETs)

Vehicular ad hoc networks (VANETs) are a key part of ITS framework, 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. Ad hoc networks do not require any kind of central unit, so the nodes in a network make communication happen across the network by collaboratively working both as a router and a node to distribute data. Ad hoc networks are created spontaneously by the participant nodes, so they are suitable for situations like rapidly changing network topology or high setting cost of a fixed network infrastructure. Even if the fundamental characteristics of VANETs are similar to MANETs, they have differences in the details. The ad hoc network is created by vehicles only, instead of any mobile devices. Thus, instead of moving in random directions with low or limited speed, vehicles could move in the same or similar paths with higher speeds. Additionally, they could have more operational and power capabilities than mobile devices. These properties of vehicles make VANETs are suitable for different kinds of applications, such as traffic information systems, road transportation emergency services, and on-the-road services. They use VANET communication in order to share knowledge about traffic with vehicles on the road, which include traffic status reports, road safety

warnings, and advertisements. Researchers are actively working on the development of such applications [1].

Inherent characteristics of VANETs bring some security challenges [2]. 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 [3], 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. They propose different ways of calculation of trust values by offering new formulas [4]. Decentralized, self-organized, autonomous, and highly dynamic topology of ad hoc networks makes the trust management an optimization problem.

1.2. Trust Management in VANETs

Trust is a concept that is derived from the social sciences and is used in information technology, specifically information security. It shows the confidence level of the trustor on the actions of the trustee. Units of information security systems, such as computer programs, algorithms, and intelligent devices become both trustors and trustees in order to establish trust management systems. Vehicles in VANETs use trust values about other vehicles in the network and make automated decisions to keep the security at an acceptable level.

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 occurring on the road such as traffic accidents,

road maintenance in order to increase traffic safety. The solution to the trust management problem must handle this safety critical dynamicity.

The proposed solutions in the literature 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 network topology environment [5]. Such optimization problems that change over time in a dynamic environment are called dynamic optimization problems (DOPs) [6], 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. Existing researches either do not take into account the DOP characteristic of VANET trust management [7] or employ a very limited dynamic approach [8], thus they can not handle the overall dynamicity of VANET as time goes on.

1.3. Dynamic Optimization

Nature-inspired optimization algorithms (NIOA) 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. They find the best or most optimal solution in the complex and large search space of an optimization problem that has too many variables for traditional algorithms, using a metaheuristic or stochastic approach. Evolutionary computation (EC) is one of the nature-inspired optimization algorithms that uses some principles of biology which are population to model a set of candidate solutions, generation to update solutions and have new populations, natural selection to select better solutions and pass their features to the next generation, mutation to introduce small changes to the population randomly, and fitness to track the success of each individual in the population. Using EC techniques to solve DOPs is named evolutionary dynamic optimization (EDO) [6]. Such EDO techniques have already been employed to solve some problems in ad hoc networks [9], but they have not been employed neither for trust management problems nor for VANET domain. In addition, 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

between the research area and real-world applications [6]. This research mainly focuses on the dynamic trust management model problem in VANETs and addresses this problem by using EDO techniques to find the trust calculation formula automatically.

1.4. Scope Of The Thesis

The complex and dynamic properties of VANETs cause finding the best formula for a trust management model harder. A lot of different evidences from the network should be analyzed and appropriate ones should be selected in order to model a trust formula that best represents the current situation of VANET. In addition to it, the dynamically changing properties of the network should be tracked and the trust formula should be changed according to changes. In according to these constraints, we propose to investigate the use of EC and EDO techniques to search the large space of trust formulas instead of developing statically constructed trust formulas. EC algorithms require fewer a priori assumptions about the problem at hand [10]. 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 [10]. EDO techniques provides such mechanisms to track the change of the best trust formula in the network over time. These characteristics of EC and EDO are among the main motivations behind using EC and EDO in this thesis.

In this thesis, an EDO based dynamic trust management model is proposed to evaluate the trustworthiness of both vehicles and messages sent by these vehicles in VANETs, to be able to distinguish the attacker vehicles from benign ones, where attackers send bogus information to the network. The previous studies in the literature generally employ statically defined trust formulas with a limited set of trust evidences for evaluating node [11] or data trust [12] and change the coefficients of parameters in such formulas to deal with dynamicity [13]. On the other hand, the proposed model generates a trust formula automatically in order to evaluate trust values by taking into account much more trust evidences than the existing studies in the literature [14, 15]. Genetic programming (GP), which is a subset of EC that evolves programs, 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 of the network over time. Furthermore, real-world application opportunities in VANETs are investigated and the proposed model is run on a traffic model taken from the movements of vehicles in real-world to help the transition of EDO research area from a theoretical view to real-world application view. To the best of our knowledge, there is no such study that automatically generates trust formulas and adapts to the dynamically changing environment for managing trust in VANETs.

1.5. Contributions

The main contributions of this thesis could be summarized as follows:

- The use of evolutionary computation techniques, specifically genetic programming, is explored to distinguish bogus information from legitimate messages and attackers from benign vehicles as well using an automatically generated trust calculation formula rather than a predefined static one. The simulation results show that GP could evolve effective trust 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 existing studies [16] in the literature, a broader set of trust evidences is given to the proposed model and the ones that best represent the current situation of the network for trust calculation are selected by GP.

- The use of evolutionary dynamic optimization techniques is explored for developing a dynamic trust management model in VANETs. The simulation 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

defines the trust management problem from the dynamic optimization problem point of view and solves the DOP in VANETs using EDO techniques.

- The opportunities of real-world applications in VANETs are investigated and the proposed model is run on a real-world traffic model to reduce the gap between the theoretical EDO research and applications of real-world DOPs.

1.6. Organization

The organization of the thesis is as follows:

- Chapter 1. presents our motivation, contributions, and the scope of the thesis.
- Chapter 2. provides a background overview about the related fields to the thesis regarding to vehicular ad hoc networks. The natural properties of VANETs bring some new security challenges to them by making them more vulnerable to attacks than other networks. These security issues of VANETs and the attacks on VANETs are presented in this chapter. Different aspects of trust establishment and management in VANETs are also defined.
- Chapter 3. provides a background overview about the related fields to the thesis regarding to evolutionary computation and evolutionary dynamic optimization. Furthermore, genetic programming, which is a subfield of evolutionary computation, is described along with different nature-inspired optimization algorithms.
- Chapter 4. gives a summary about 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. The complexities and difficulties of trust management in VANETs are reviewed and its open issues are outlined in this chapter.
- Chapter 5. introduces the main focus of this thesis, which is the proposed evolutionary dynamic optimization method for building a dynamic trust management system in

VANETs. All properties of trust management systems which are covered by the proposed method and trust evidences used to build the trust management are described.

- Chapter 6. demonstrates the experimental results of the proposed method. The ability of evolved trust formulas to detect bogus information messages is shown on simulated networks. Moreover, a real-world application case study of the proposed method is presented and its experimental results are discussed in this chapter. The limitations of the proposed approach and the possible future research directions for the problem are also discussed in this chapter.

- Chapter 7. states the summary of the thesis and possible future directions. The suitability of the proposed dynamic trust management model to VANETs is discussed.

2. VEHICULAR AD HOC NETWORKS

The purpose of this chapter is to provide background information about the research areas in which this research is conducted, regarding to vehicular ad hoc networks. Ad hoc networks are given in Section 2.1., vehicular ad hoc networks are presented in Section 2.2., and intelligent vehicular ad hoc networks are described in Section 2.3.. The security challenges of vehicular ad hoc networks are detailed in Section 2.4., attacks on vehicular ad hoc networks are given in Section 2.5., and trust establishment and management in vehicular ad hoc networks are described in Section 2.6..

2.1. Ad Hoc Networks

Ad hoc networks, also known as wireless ad hoc networks, are a form of wireless networks which do not require a preexisting infrastructure, like routers or access points. Each member of the network, named node, can enter to the network anytime and can exit from it freely. Because of the lack of infrastructure, each node can behave both as a router and a node. Hence, nodes can communicate directly with each other even when they are not in their transmission ranges. They rely on other nodes in the network for forwarding their packets. Communication can be made in different ways, such as forwarding packets to nodes along a path from their source node to their destination node, or forwarding packets to all nodes in the network. Nodes in an ad hoc network are also free to move at any speed in any direction, independently from each other, thus their neighbour nodes which are in the direct communication range of themselves change frequently. Forwarding becomes a significant point in such a situation to maintain the flow of information in the network continuously. These properties make the ad hoc networks are self-configuring, decentralized, and dynamic networks. They are suitable for various applications where a network with a fixed infrastructure can not be established because of time and cost constraints such as natural disasters, thanks to their decentralized natures. Their fast deployment and self-configuring capabilities make ad hoc networks also favorable for rapidly changing topologies such as military operations and road traffics.

Ad hoc networks can be classified into different categories according to the types of nodes that form the network and the fields of their applications. Mobile ad hoc network is the most common subtype of ad hoc networks in which mobile devices connect spontaneously as nodes of the network. The mobile nodes move dynamically, thus the connections between nodes are changing continuously, making the ad hoc network a self-organizing, infrastructureless one. It is shown in [17] that the mobility of nodes in an ad hoc network improves the performance of the network by increasing the network capacity compared to a fixed ad hoc network. This significantly increased the research interest in MANETs and other similar ad hoc networks over the years. Other categories include smartphone ad hoc networks which rely on hardware in smartphones without using a cellular network, wireless mesh networks where each node is fully connected to every other node with low mobility, flying ad hoc networks (FANETs) in which the nodes are unmanned aerial vehicles with great mobility and capability to complete a military mission and task collaboratively, wireless sensor networks where sensors connect to each other to gather a large scale data, and so on.

2.2. Vehicular Ad Hoc Networks

Vehicular ad hoc networks are introduced by applying the principles of MANETs into the vehicular field and indicate that vehicles take part in the network as nodes. These vehicles are equipped with the required devices to be able to have communication capabilities. They can enter an ad hoc network while they move along the road and they can start to communicate with other vehicles on the same road instantly. Different from the mobile nodes of MANETs, vehicles move at high speeds and they follow a path on paved roads rather than moving in random directions. The primary aim of VANET applications is achieving a safe transportation in traffic by preventing collisions and accidents. Secondary objectives of them may include providing more effective and efficient navigation by notifying traffic jams, closed roads, etc. In order to accomplish the objectives, vehicles in the ad hoc network send messages including information about safety warnings, traffic statuses, and road services to other vehicles in their communication range. Upon the distribution of information in the network, vehicles can take action to avoid accidents and traffic congestion.

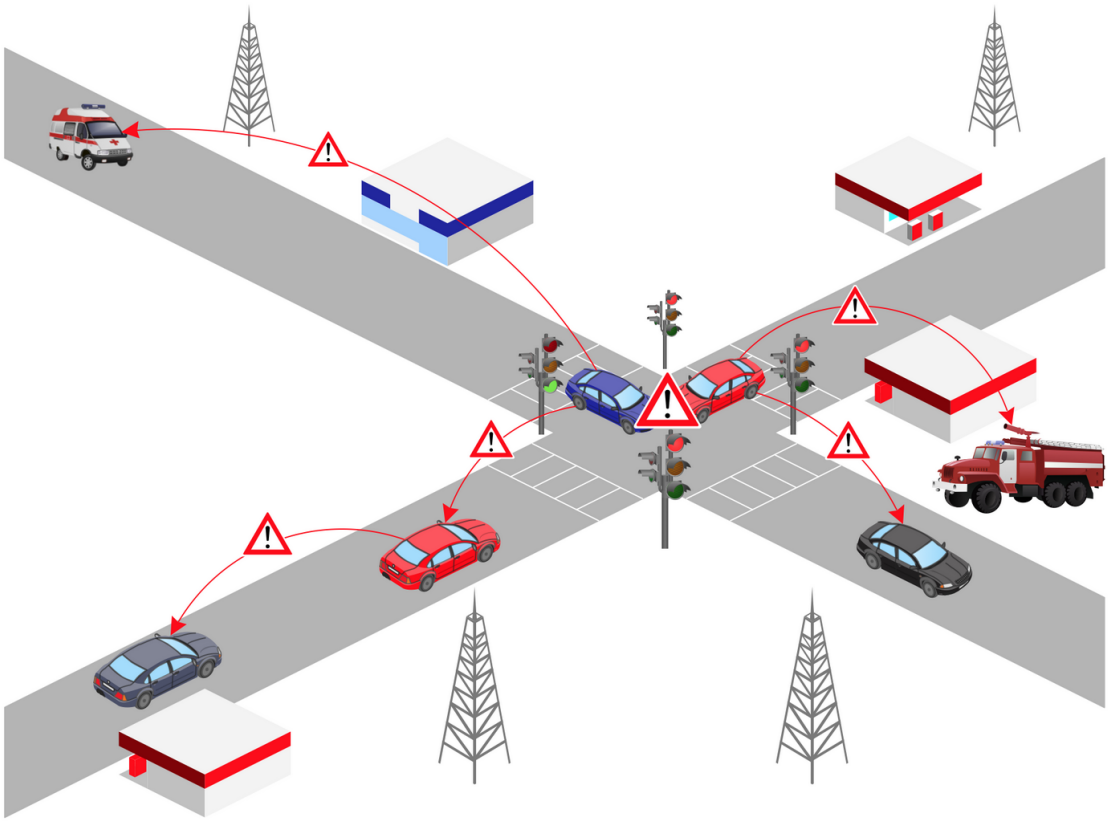


Figure 2.1 Example of a Vehicular Ad Hoc Network (VANET)

Vehicular ad hoc networks could optionally include fixed equipment that are placed along the road, named as road side units (RSU). In this case, the type of communication turns into a vehicle-to-infrastructure (V2I) rather than being only a vehicle-to-vehicle (V2V) communication. Additionally, this network could be expanded to a vehicle-to-everything (V2X) communication, adding any other entities such as pedestrians. All of these communication types have research interests to a certain extent in order to increase road safety and improve traffic efficiency by developing different VANET applications. Communication types of vehicular ad hoc networks apart from the vehicle-to-vehicle are beyond the scope of the study in this thesis. An example of a typical vehicular ad hoc network is illustrated in Figure 2.1 [18].

2.3. Intelligent Vehicular Ad Hoc Networks

Artificial intelligence applications that are executed on the devices of vehicles provide intelligent behaviour capabilities to vehicles during safety critical incidents in traffic such as collisions and accidents. Networks of vehicles with such capabilities are named as intelligent vehicular ad hoc networks (InVANETs). A variety of applications could be applied in InVANETs ranging from safety and efficiency to infotainment, providing alerts and/or warnings to the driver, finding an optimal navigation path for minimizing the intensity of traffic, locating landmarks in a newly arrived city, etc. Furthermore, vehicles could process the information shared on the network in near real time and automatically take actions according to the data aiming to increase the safety of traffic. Hence, the ad hoc network could evolve into an intelligent vehicular ad hoc network of autonomous vehicles.

2.4. Security Challenges of VANETs

Vehicular ad hoc networks have a dynamically changing topology by the nature of ad hoc networks, as they allow vehicles can enter to the network, move in any direction, and exit from the network, independently from any other vehicle. They generally do not implement any network access control policy, so any kind of vehicle can enter to the network without encountering with authentication and authorization. There is also no control over the behaviour of the vehicles in the network, thus any vehicle could either behave honestly or behave improperly. Benign vehicles contribute to the VANET according to its requirements and objectives. They send proper messages about their observations and share the information taken from other vehicles in the network. Malicious vehicles prioritize their own benefits over the VANET objectives. They either only send and share messages about their own interests selfishly rather than sharing all information or totally aim to disrupt the communication in the network by attacks. Moreover, the vehicles that priorly benign ones could be compromised by malicious vehicles and their behaviour could turn into detrimental. VANETs are vulnerable to such harmful activities and attacks, so they need to implement some lightweight protocols to be able to secure and safe against such attacks.

The natural properties of VANETs bring some new security challenges to them, so they should be taken into account while proposing new secure and safe protocols. These security issues of VANETs are listed as follows.

2.4.1. Privacy

There is a trade-off between privacy and security [19]. Protocols that are using some information from vehicles and drivers in order to increase the security level of VANETs, should respect the privacy of drivers which may not want to share their private information. This concern makes it difficult to establish a secure communication while protecting the privacy of users at the same time.

2.4.2. Scalability

The number of vehicles globally on the road is growing rapidly, which is expected to reach 2 billion in 2030, and it is also expected that there will be 863 million vehicles connected to VANETs on the road worldwide in 2035 [20]. This brings the need of scalable VANETs in order to manage high loads of communication [21].

2.4.3. Mobility

Vehicles in VANETs are much faster than nodes in any other ad hoc networks. The high speeds of vehicles cause the topology of VANETs changes very rapidly. As a result, the communication between vehicles generally lasts for a limited time in VANETs. In addition, vehicles driving in opposite directions could make only one-time interaction, and then could not communicate with each other again.

2.4.4. Real-time Communication

Most of the safety-related applications in VANETs have the requirement of real-time communication. The lack of real-time communication could cause safety-related incidents such as traffic accidents. Attacks targeting the real-time communication should be either prevented or detected in order to preserve the safety in VANETs [22].

2.4.5. Cooperativeness

Communication in VANETs relies on the dissemination of data by vehicles which requires the cooperativeness of vehicles. Uncooperative vehicles perform bogus information attacks thus break the communication, which is a security issue that needs to be detected and prevented.

2.5. Attacks on VANETs

VANETs are the target of many attacks that could cause serious life-threatening effects by the open nature of them. Any attacker vehicle can aim to either harm the network or gain benefit from it. These attacks must be detected by security solutions to prevent their damage to the users of vehicles. A broad categorization of attacks on vehicular communication systems and their effects are presented by Sakiz and Sen [3]. The attacks on VANETs that target the integrity of data transmitted in the network are the main focus of this thesis, so other types of attacks are out of scope for this research and data integrity attacks are categorized as follows.

2.5.1. Bogus Information Attack

The success of the communication in a VANET depends on the trueness of the information shared across the network. However, malicious vehicles could send false information to their neighbours. The attackers aim to convince other vehicles to believe their messages and manipulate their movement in the road, hence they could achieve a traffic-free road along

their path. Any kind of message can be sent as bogus information such as a fake accident or traffic congestion message, so the receivers of this message could take another road if they can not identify the attack. Since there exists a variety of bogus information attacks requiring different countermeasures, they are discussed separately under different names which are message falsification attack, message spoofing/forgery attack, and message alteration attack.

2.5.2. Message Falsification Attack

The attacker vehicle sends the opposite of the messages sent by benign vehicles in a context. It tries to convince vehicles that its own message is true rather than the messages of others in the context. The context of messages could be about any situation in the road traffic, for example, an attacker could send a different message for the position of a traffic accident while benign vehicles are sending accident messages for that coordinate. In a situation in which a vehicle has only two neighbours and one of them is an attacker, deciding which one of the two conflicting messages is true is harder. Undoubtedly, sending false information about a safety-related context is more dangerous than others.

2.5.3. Message Spoofing/Forgery Attack

Malicious vehicle generates a fake message about an inexistent context as if it is just occurred and sends the message to other vehicles. It aims to gain a benefit on the road, such as reaching a traffic-free transportation while sending traffic jam messages to its neighbours. It is the only vehicle that is sending a message in the context at the beginning, but if its neighbours can not detect it and begin to spread the fake message, the network could easily be harmed by the attacker.

2.5.4. Message Alteration Attack

Vehicles modify the content of the received messages while forwarding them to other vehicles in this attack scenario. The receiver of the altered message could classify the original

sender of the message as a bogus information attacker. The messages sent in the network are assumed as could not be modified but only could be extended by the receiver vehicles in this study.

2.5.5. Sybil or Impersonation Attack

Vehicles change their identities while sending messages to other vehicles in the network in these attack scenarios to introduce themselves as different vehicles, thus making the attackers are undistinguishable from benign vehicles. Additionally, they could generate more than one identity and send messages with different identities, so these messages seem as coming from more than one vehicle. Sybil and impersonation attacks are out of scope for this study since the identities are assumed as could not be changed by the vehicles themselves.

2.5.6. Message Replay Attack

Attacker vehicle stores the incoming messages and sends them in another time as if they are just sent by the original sender. These replayed messages could easily be rejected by the vehicles using a timestamp value in the message and defining a short lifetime for the messages in the network.

2.6. Trust Establishment and Management in VANETs

Securing VANETs is not a straightforward task because of the aforementioned challenges and attacks. Different VANET environments require to implement different security measures. These security measures could be categorised into one of the three phases in a security lifecycle similar to other networks, which are prevention, detection, and response [23]. Prevention mechanisms are used to secure the network from external attackers. Authentication techniques are most common and widely used ways to prevent attacks in wired networks, such as symmetric and public key systems. The lack of a central authority in ad hoc networks makes the implementation of authentication techniques is a

challenging issue. Although several authentication mechanisms are developed for MANETs, the unique features of VANETs such as highly dynamic network topology, impose a great challenge to implement an authentication scheme [24]. Detection and response mechanisms complement the prevention mechanisms. Anomaly or misuse detection techniques could be used to detect all the intrusive activities in ad hoc networks, as in wired networks [25]. Both techniques have their own capabilities and limitations, so the combination of these could improve the intrusion detection in ad hoc networks. The method of response is another important phase of the security lifecycle, which is activated when an intrusion or an attack is detected. Each node in an ad hoc network is responsible to respond actively to any attack and mitigate the effects of attacks. Trust establishment and management is one of the detection and response mechanisms used in ad hoc networks. Vehicles that communicate in VANETs establish a trust relationship between them and share their trusted and untrusted parties with other vehicles along the VANET. This process enables the detection of malicious vehicles actively and avoiding any communication with the attackers.

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 [26–30]. Dynamicity, context-dependency, subjectivity, asymmetry, and incomplete/partial transitivity are described as trust properties. Nonetheless, none of the proposed approaches for VANETs covers all trust properties [27].

2.6.1. Dynamicity

First trust property is dynamicity. Trust is a dynamic concept, not static [31]. Trust establishment should be based on local information and temporally in VANETs, because the gathered information can change rapidly due to the highly dynamic mobility of vehicles [32]. The dynamic and uncertain VANET environment makes trust useful for vehicles to achieve their goals. If vehicles move in a predictable environment and do not need to interact with other vehicles, using trust would be meaningless.

2.6.2. Context-dependency

Secondly, trust is context dependent [33]. A vehicle can trust another vehicle only within a context in VANETs. Vehicles could behave differently in different contexts, such as sending genuine messages in a non-safety-related context but sending counterfeit messages in a safety-related context. As a result, a vehicle may trust another vehicle in one context but not in another context, such as trust in sending message as a source versus trust in forwarding messages of others.

2.6.3. Subjectivity

Third, trust is subjective [34]. In VANET environments, subjective trust means that any two vehicles may have a different degree of trust to the same vehicle even if the trustworthiness of the vehicle remains constant, because they have different experiences with the same vehicle due to dynamically changing network topology.

2.6.4. Asymmetry

Fourth, trust is asymmetric [31]. Trust is not symmetric because the knowledge of vehicle A about vehicle B may not be the same as the knowledge of vehicle B about vehicle A. These vehicles trust each other at different levels.

2.6.5. Incomplete/Partial transitivity

Lastly, trust is partially transitive [35]. Trust is not fully transitive because each vehicle establishes a trust relationship from its own perspective rather than trusting the trustees of its own trustee. A vehicle evaluates the recommendation of its own trustee about a third vehicle and then decides to either trust or not trust the third vehicle that is recommended. This makes the transitivity of trust between two vehicles only to some extent, which is an incomplete transitivity of trust.

2.6.6. Trust Management Properties

In highly dynamic and distributed environments such as VANETs, trust management should be fully decentralized [29]. 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 [27]. 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 [29]. 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 [29]. Moreover, the possibility of uncooperative vehicles to enter VANETs freely should also be taken into account in developing a trust management model [27, 29].

2.6.7. Trust Models

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 trust models, data-oriented trust models, and hybrid trust models [28, 29]. **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**. It is stated in [1] that the trustworthiness of the data is more useful than the trustworthiness of the nodes. **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.

3. EVOLUTIONARY COMPUTATION

The purpose of this chapter is to provide background information about the research areas in which this research is conducted, regarding to evolutionary computation. The nature-inspired optimization algorithms are presented in Section 3.1., swarm intelligence is given in Section 3.2., evolutionary computation and specifically genetic programming are described in Section 3.3.. Other bio-inspired algorithms are given in Section 3.4. and evolutionary dynamic optimization methodology is detailed in Section 3.5..

3.1. Nature-Inspired Optimization Algorithms

Nature has physical laws like gravitation and always changing biological conditions. All organisms have to adapt and optimize themselves to these changing conditions to be able to survive. They perform some actions such as searching for food and maintaining their species. Nature-inspired optimization algorithms are a category of algorithms that are derived from these natural phenomena. They use a stochastic approach while searching an optimum solution to complex and high-dimensional problems which have a large search space. All nature-inspired optimization algorithms maintain two kinds of search to reach the best solution, a global search which is named **exploration** and a local search which is named **exploitation**. These algorithms are classified into bio-inspired algorithms and physics-based algorithms. The bio-inspired algorithms are further classified into evolutionary algorithms (EA), swarm intelligence (SI), and other bio-inspired algorithms in this thesis. This research mainly focuses on the evolutionary algorithms subfield of nature-inspired optimization algorithms, which are studied under the evolutionary computation terminology. Having said that, some of the algorithms apart from EA are also mentioned in Section 4.2. while reviewing the nature-inspired optimization algorithms that are used in ad hoc networks, thus a brief overview of them is presented in this section before giving the details of evolutionary computation. A taxonomy of nature-inspired optimization algorithms is shown in Figure 3.1, presenting only the algorithms mentioned in this thesis for the sake of simplicity.

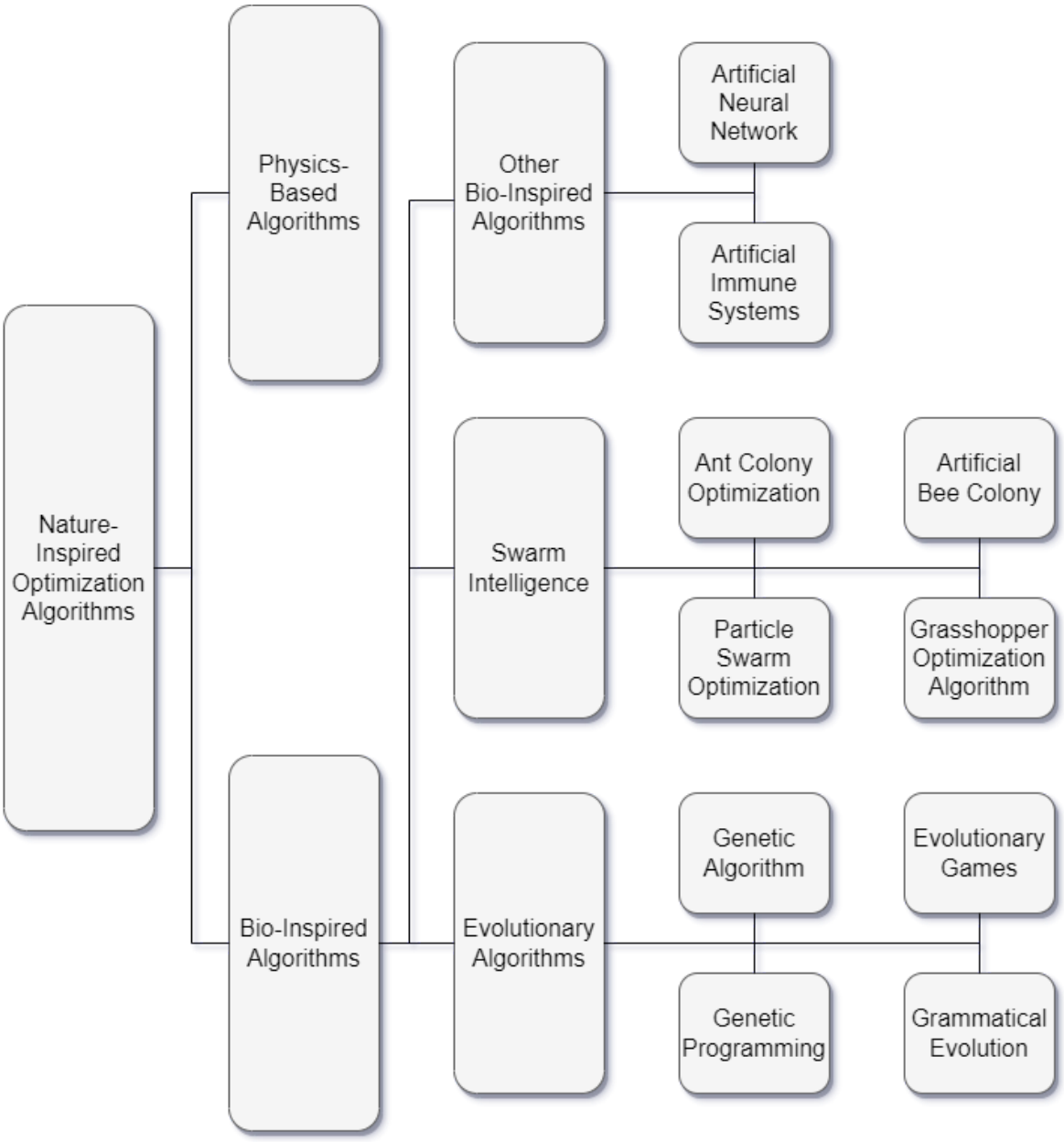


Figure 3.1 Taxonomy of nature-inspired optimization algorithms

3.2. Swarm Intelligence

Swarm intelligence is the main subfield of bio-inspired algorithms inspired by the collective behaviour of individual biological systems such as ants, bees, birds, and fishes. In nature, each individual in a population interacts and communicates with others while acting on its local environment, so this behaviour contributes to maintaining the lifecycle of the

population, for instance, ant and bee colonies, flocks of birds, and fish shoals. This collective behaviour is modeled by a decentralized, self-organized artificial population of agents in computing systems. The local and random interactions between agents without the centralized control structure bring out the intelligent global behaviour which have problem solving capabilities that individual agents can not have.

3.2.1. Ant Colony Optimization

Ant colony optimization (ACO) algorithm is a probabilistic problem solving technique that models the actions of an ant colony [36]. Artificial ants search optimal solutions by finding better paths through graphs while moving through all possible solutions, like natural ants explore the best path between the nest of their colony and the food source. Similarly to laying down pheromones by natural ants to inform others about food sources while moving back to the nest, the simulated artificial ants put a record about the position of them and the quality of the solution they had found, hence ants in later iterations of the simulation could search better solutions using previous artificial pheromone records.

3.2.2. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based global optimization algorithm that searches the optimum solution by trying to improve candidate solutions iteratively through a multidimensional solution space [37]. Each candidate solution in a population, named particle, moves through the search space of the problem in each iteration using its local past best known position and the global best known position which is gathered by communicating with other particles, providing a movement of the swarm toward the global optimum. This behaviour represents the movement of biological organisms in swarms like a bird flock and a fish shoal.

3.2.3. Artificial Bee Colony

Artificial bee colony (ABC) algorithm is a multidimensional and multimodal optimization algorithm that simulates the intelligent foraging behaviour of honey bee swarm [38]. Three groups of bees perform three different tasks to reach better food sources in foraging, thus three groups of artificial bees are included in the algorithm model. The first group examines food sources and collects information about them, represented by artificial bees that calculate the quality of candidate solutions. Second group evaluates the findings of the first group and chooses a food source, modeled by artificial bees that choose the best candidate solution so far based on the local search of the first group. The last group searches new food sources, simulated by artificial bees that discover new candidate solutions randomly in the search space of the problem. This intelligent foraging behaviour of honey bee swarm is well suited for modeling the exploration and exploitation capability of an optimization algorithm.

3.2.4. Grasshopper Optimization Algorithm

Grasshopper optimization algorithm (GOA) is a recent swarm intelligence algorithm that mimics the foraging behaviour and interaction of grasshopper swarms in nature [39]. Each grasshopper corresponds to a candidate solution in the population and the movement of the natural swarm for searching food sources represents the searching global optimum solution of the algorithm to the optimization problem. In one of the two phases of the lifecycle of grasshoppers, the nymph does not have wings, thus it has a small step size and moves slowly. This local movement behaviour is modeled as the exploitation stage of the algorithm. The adult grasshopper has the long-range movement ability by sudden jumps in the second phase of the lifecycle of grasshoppers, which is modeled as the exploration stage of the algorithm.

An introduction to some swarm intelligence and bio-inspired algorithms is given in this study due to their usage in solutions to optimization problems in ad hoc networks. Other nature-inspired optimization algorithms are also exist in the literature apart from the algorithms that are mentioned in this thesis. They are left out of the scope for the sake

of simplicity because these are used less or not at all for optimization in the context of ad hoc networks.

3.3. Evolutionary Computation

Evolutionary computation is the main subfield of bio-inspired algorithms inspired by biological evolution. In nature, some characteristics of biological populations change over generations and they are passed on from parents to their offspring via genes, which is called as inheritance. The genetic drift process of biological evolution changes an existing gene of individuals by random chance, so the frequency of gene variants in populations are changed randomly, which is called as **mutation**. The **natural selection** process of biological evolution determines individuals of the population which have appropriate characteristics to fit to the conditions of the environment and thus to survive in nature. These individuals produce more offsprings, so individuals of the next generation have increased fitness, which is called as **survival of the fittest**. These processes of biological evolution are simulated by computational systems to build an artificial evolutionary system. This system includes a population of individuals which are candidate solutions to a problem, a fitness criterion to calculate the fitness level of each individual to the problem, and evolutionary operators to produce offsprings from parents which will become the population of the next generation. In evolutionary computation, the initial population is generated randomly and updated iteratively using evolutionary operators such as mutation and recombination. Fitness values of the individuals are evaluated using the fitness function of the problem in each iteration. Individuals that have higher fitness values are selected as parents using a selection operator inspired by survival of the fittest and natural selection. Individuals of the next generation are produced by using evolutionary operators on parents. This simulated evolution process evolves the population to increase the fitness of it according to the chosen fitness function that models the problem and converges towards a nearly optimal population of individuals. In computer science, evolutionary computation techniques are a popular research area thanks to their capability of producing highly optimized solutions in a wide range of problem domains. The evolutionary computation research area is mainly

classified into three categories: evolution strategy (ES), evolutionary programming (EP), and evolutionary algorithms. This research mainly focuses on the evolutionary algorithms subfield of evolutionary computation, so other subfields are not described in this section and are not shown in Figure 3.1 for the sake of simplicity.

3.3.1. Evolutionary Algorithms

Evolutionary algorithms is a subfield of evolutionary computation, which comprises of population-based metaheuristic optimization algorithms. In evolutionary algorithms, individuals of a population correspond to candidate solutions to an optimization problem and the quality of the solutions to solve the problem is determined by evaluating the solution using the fitness function which specifies the environment of the problem. EA algorithms use the mechanisms of EC which are natural selection, survival of the fittest, reproduction, recombination, and mutation, that are all inspired by biological evolution. The application of these mechanisms iteratively evolves the population into a nearly optimal solution to the problem.

Evolutionary algorithms generally do not make any assumption about the optimization problem and its fitness landscape, which is the distribution of all fitness values of all candidate solutions evaluated against the fitness function. They use stochastic methods to find an approximating solution to problems, so evolutionary algorithms generally perform well on all types of optimization problems. The pseudocode of a generic single-objective evolutionary algorithm is given in Alg. 1. It starts with an initial population, in which the individuals are randomly generated across the search space. The size of the population could be determined arbitrarily for the specific optimization problem. Each individual in the population is evaluated using a fitness function that shows how well is the individual performed to solve the problem, which is the fitness value of the individual. After the fitness values of all individuals are evaluated, some individuals are selected to survive in the problem environment and have the chance of being reproductive for the individuals of the next generation. These selected parent individuals breed new offspring individuals for the

next generation using the genetic material of them. Through the recombination of successful parts of multiple genetic materials and the random mutation of the genetic material, ideally the next generation population has more successful individuals to solve the optimization problem. These newly bred offspring individuals are replaced the least-fit individuals of the current population in order to create the next population. This whole process continues iteratively until automatically or manually terminated. In order to terminate the evolution automatically, the algorithm must find the most ideal and optimal solution to the optimization problem, which is generally can not be applicable. Therefore, the evolution could be terminated either when a nearly optimal solution is found or at a predefined generation number. This criterion of termination is determined according the problem and when it is satisfied, the most fit individual found so far is chosen as the solution to the optimization problem. In this algorithm, the selection operator increases the quality of the population and recombination and mutation operators increase the diversity of the population in order to reach more optimal solutions. Individuals with higher fitness values have a higher chance to be selected rather than individuals with lower fitness values. These operators use a stochastic approach, so weak individuals also have a chance to survive and become a parent. Recombination and mutation operators choose the changed pieces of individuals randomly. Evolutionary algorithms differ in the representation of individuals, the implementation of operators, and the nature of the problems they are applied to.

Algorithm 1 The pseudocode of a generic single-objective evolutionary algorithm

```
1: generate the initial population of individuals randomly
2: while a termination criterion is not satisfied do
3:     evaluate the fitness value of each individual in the population
4:     select the fittest individuals for reproduction as parents
5:     breed new individuals through recombination and mutation operators as offsprings
6:     replace the least-fit individuals of the population with new individuals
7: end while
8: return best-of-run individual as the solution to the problem
```

3.3.2. Genetic Programming

Genetic programming evolves computer programs, which are represented as tree structures [40], instead of writing computer programs by applying the human intelligence at hand [41, 42]. It is one of the population-based optimization techniques, thus evolutionary computation principles such as survival of the fittest, natural selection, recombination, and mutation are applied to evolve these tree structured computer programs. Koza describes that the genetic programming paradigm searches the solution space of computer programs in order to find the most fit candidate solution to solve different kinds of problems from a variety of fields including machine learning and artificial intelligence [42].

Complex organisms in nature gain new capabilities to be able to live and survive in nature by evolving over large periods of time with the help of natural evolution. Living organisms try to adapt to conditions of the environment in nature. The conditions of the environment specify a fitness, the success of an organism. Most successful organisms continue to live as others cannot, i.e., only the fittest organisms of the population in the environment can survive in nature, which is named as survival of the fittest in natural evolution. These fittest organisms have more chance to produce new organisms than others as the consequences of natural selection. The reproduction, which combines the genetic material of parent organisms into an offspring organism, and the mutation, which is an alteration in the genetic material of an organism, provide a way to increase the adaptability of organisms in the population to the conditions of the environment. These processes of natural evolution are applied into the programming domain, hereby it enables creating more complex computer programs automatically than humans can write to solve complex problems.

In genetic programming, an initial population of computer programs as individuals are created randomly. Each computer program tries to solve the problem as a candidate solution, which is an analogue of organisms trying to survive in an environment. Success level of each program is evaluated according to its capability of solving the problem and result of the evaluation shows the fitness value of the program. A selection mechanism determines which programs in the population can survive to the next generation using probability values

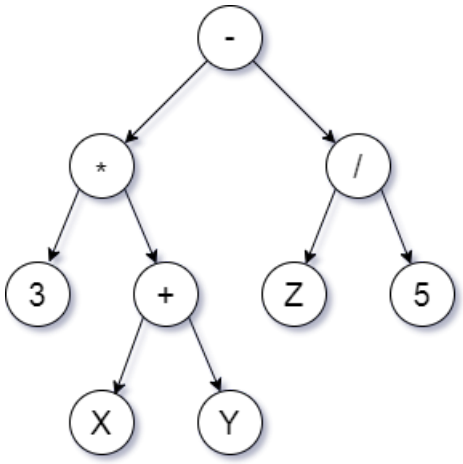


Figure 3.2 An example representation of an individual in genetic programming

proportionate to their fitness values. Selected programs play the role of parents as new offspring programs are created by recombining some parts of the parent programs randomly. This evolution process of the population of programs continues over a number of generations.

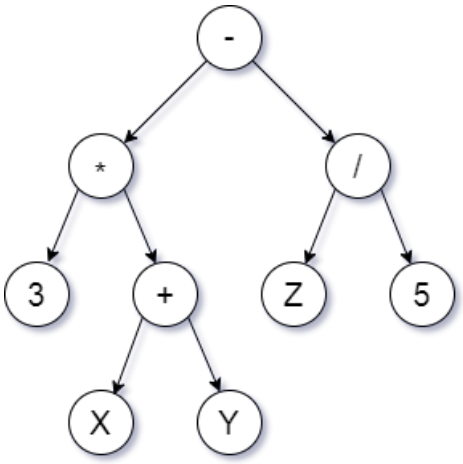
Different genetic representations are used in different evolutionary computation methods. Individuals of the population are represented as hierarchical tree structures in genetic programming. These hierarchical tree structures could represent several types of functional computer programs, such as boolean-valued, integer-valued, real-valued, vector-valued, and complex-valued, with preferred properties for many different problems in artificial intelligence [43]. In order to evolve and construct a mathematical formula to be able to calculate the trust values, an appropriate tree structure is chosen in this study. Terminal nodes of the tree include the input variables of the problem domain and constant values. Arithmetic operations, mathematical functions, logical operations, and problem-specific functions are used in non-terminal nodes of the tree. In-order traversal of the tree gives the formula which is the computer program represented by the individual. Figure 3.2 shows an example representation of an individual including simple arithmetic operations, three inputs, and some numeric constants. The mathematical formula which the example tree represents is shown in Eq. 1.

$$[3 \times (X + Y)] - (Z/5) \tag{1}$$

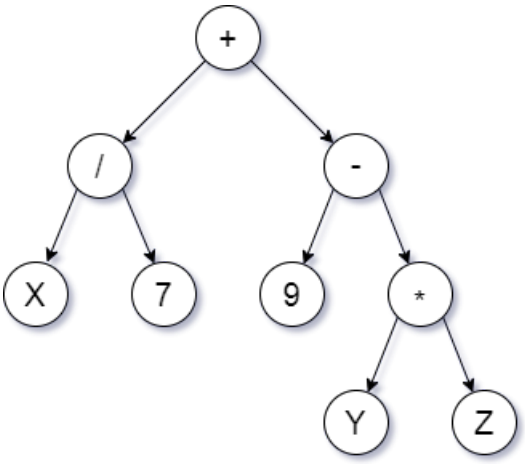
Each function is evaluated and the result value shows the success level of the corresponding individual about solving the problem, which is named as the fitness value. The method of evaluation is called as the **fitness function** and it varies according to the problem definition. The individuals which have higher fitness value are more closer to solving the problem than other individuals. These fitter individuals have a higher chance to be selected as parents by the selection method used in genetic programming. New offspring individuals are created as the population of the next generation using these parents.

3.3.2.1. Selection is the method which probabilistically selects some individuals from the population of the current generation as parents. Even if the more successful individuals have a higher chance of getting selected, the less successful ones could also be selected due to the probabilistically selection method [44]. There exist several selection methods in genetic programming that perform better for different problems: tournament selection, fitness proportionate selection, rank selection, lexicase selection [45], and others. The most commonly used one is tournament selection, that runs several tournaments among randomly chosen individuals and selects the winner of each tournament, which is the participant of the tournament with the best fitness. The participant size of the tournament could be adjusted according to the problem, then it affects the probability of selection of weak individuals [46]. If the tournament size is specified as 1, the selection method will become a random selection. Fitness proportionate selection, i.e., roulette wheel selection, assigns a probability of selection to each individual proportional to its own fitness level, which is calculated by dividing the fitness value of an individual by the total fitness value of the population, and then selects randomly. Rather than depending directly on the fitness value for the selection probability, the fitness values of individuals within the population are ranked from higher to weaker, and then the selection probabilities are assigned according to the rank of the individuals in the rank selection [47, 48].

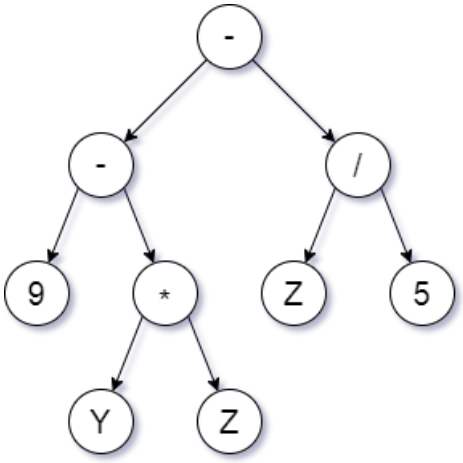
3.3.2.2. Crossover , i.e., recombination, is the reproduction process between two selected parent individuals of the population to create two new offspring individuals for



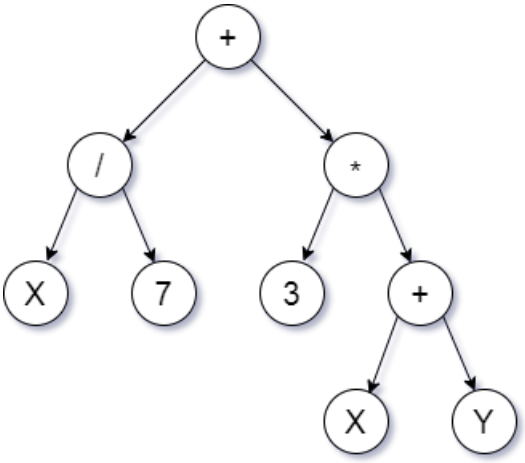
(a) First parent individual



(b) Second parent individual



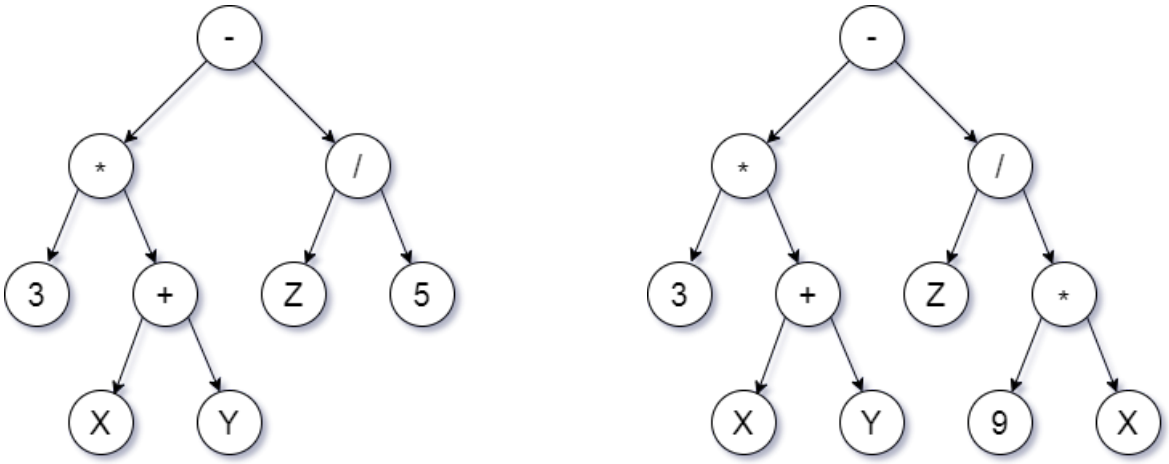
(c) First child individual



(d) Second child individual

Figure 3.3 An example of crossover operation of two parent individuals in genetic programming

the population of the next generation. These offsprings carry a combination of randomly selected parts of the parents. The offsprings which have acquired successful parts of both parents become more effective in solving the problem. In the crossover operation, a subtree is randomly selected as the crossover point of each parent. These two subtrees are then exchanged and replaced with each other. Figure 3.3 shows an example of crossover operation of two parent individuals. The crossover points of the parents are * and -, respectively. The result of exchanging subtrees at crossover points is the two offsprings that are shown in Figure 3.3.



(a) Before the mutation

(b) After the mutation

Figure 3.4 An example of mutation operation on an individual in genetic programming

3.3.2.3. Mutation process alters a randomly selected part of an individual of the population to create a new individual for the population of the next generation. This individual gains different abilities while carrying most of the genetic material of the parent individual. This process introduces diversity to the population at a level to be able to search the areas of the solution space which the population of the current generation does not cover yet. In the mutation operation, a subtree is randomly selected as the mutation point of an individual. This subtree is then removed and replaced with a randomly generated subtree. Figure 3.4 shows an example of mutation operation on an individual. The mutation point of the individual is the node with the value 5. This subtree of only one node is replaced by another subtree of three nodes, which is equivalent to $9 \times X$.

3.3.2.4. Elitism operation selects the best individual of the current generation in terms of fitness value and allows it to survive in the next generation by moving it into the new population without changing. This is generally employed to avoid extinction of the best solution known so far. Individuals of the next population are generated when the crossover, mutation, and elitism operations are performed on the current population. The new population replaces the old population as the next generation. The fitness value of

each individual in the next generation is then calculated and the whole evolution process is repeated over many generations until an acceptable enough solution to the problem is found.

3.3.2.5. Replacement strategy specifies how the offspring individuals are substituted for parent individuals in the population. Generally, two main strategy are used in GP to replace individuals: the generational replacement and the steady state replacement. The whole newly generated offspring population is replaced with the entire parent generation in the former strategy. In the latter, only one offspring individual is generated and replaced with the least-fit parent individual in each generation.

3.3.3. Genetic Algorithm

Genetic algorithm (GA) is a type of evolutionary algorithm, in which the individuals of populations are represented as strings of numbers and are evolved to better solutions by applying evolutionary operators to solve optimization and search problems [49–51]. A standard genetic representation of each individual in the genetic algorithm is an array of 0s and 1s, i.e., a bit string, but arrays of other types could also be used. This representation shows the set of properties of each candidate solution like chromosomes or genotype in biology, so they can be altered by crossover and mutation operations. Simple genetic representations have fixed length arrays and enable easier crossover operation which selects the same portion of chromosomes of both parents and swaps them to recombinate new offsprings. Variable length arrays could also be used as genetic representations, but the crossover operation becomes more complex. A common mutation operator flips the randomly chosen bits in a bit string representation of chromosomes according to a probability.

Genetic algorithm has variants of the standard process in different areas of methodology, such as chromosome representation, evolutionary operators, and adaptation. Genetic representations other than the bit string, like an array of integers and real numbers or different data types, are some variants of the genetic algorithm. A main variant of the genetic

algorithm is created using an additional evolutionary operator, the elitism operator. It carries over the best individuals of the current generation to the next generation, without altering its representation of chromosomes. Its main aim is to keep the quality of the solution evolved so far and prevent a decrease in the fitness of the population. Another variant of genetic algorithm uses adaptive parameters for the probabilities of crossover and mutation operators in order to maintain diversity of the population in addition to the convergence capacity.

3.3.4. Evolutionary Games

Evolutionary games (EG) are game theory applications which use evolutionary algorithm methodologies to provide an optimal or near-optimal solution for a fully dynamic game model [52]. It focuses on the dynamics of strategy change in the population, which is formed by players. The fitness function of evolutionary games provides that more successful strategies have an increased probability to be selected for reproduction.

3.3.5. Grammatical Evolution

Grammatical evolution (GE) is a variation of genetic programming, in which individuals are represented by a string of integers and are mapped to programs by using Backus–Naur form grammar [53]. The context-free grammar describes the rules and they are selected according to an evolved list of integers. The search process of the grammatical evolution could be the same as in genetic algorithm. Evolved individuals are then mapped to tree-like structures. Output of the rules is a program tree, which is the same as genetic programming.

3.4. Other Bio-Inspired Algorithms

The bio-inspired algorithms other than EC and SI are described in this section.

3.4.1. Artificial Neural Network

Artificial neural network (ANN) [54] is inspired by the interconnected network of biological neurons. It is composed of a collection of interconnected units named artificial neurons, which model the connections of biological neurons. Like a biological neuron passes signals to other neurons through synapses, artificial neurons receive real-valued inputs from others, process them using weights, and send a real-valued output to other artificial neurons.

3.4.2. Artificial Immune Systems

Artificial immune systems (AIS) [55] are inspired by the learning processes and memory characteristics of natural immune systems that protect biological organisms from diseases. The natural immune systems detect pathogens that are not belong to the organism such as viruses, bacteria, worms, and cancer cells and react to them. Abstractions of the structure and function of the natural immune system into the computational system bring out the AIS, which are used in applications such as computer virus detection, anomaly detection.

3.5. Evolutionary Dynamic Optimization

Evolutionary dynamic optimization is a recent and active research area in the evolutionary computation field. The main focus of the area is adapting evolutionary optimization into dynamic environments, which is defined as using evolutionary computation techniques and other bio-inspired algorithms in order to solve dynamic optimization problems [6]. Dynamic optimization problems are a special class of dynamic problems in the literature of optimization in dynamic environments, which are solved online by an optimization algorithm while the optimization problem changes over time [56]. The difference between dynamic optimization problems and general dynamic problems is clearly stated that the optimization algorithm must take into account the dynamics of the problem and produce new solutions in order to react to changes as time goes by [57]. Optimization in dynamic environments is an important and challenging task because the conditions of many real-world optimization

problems are changing over time, so the optimal solution of the problem is also being changed over time, thus a dynamic optimization algorithm must both find an optimal solution of a dynamic problem and track the change of the optimal solution of the problem over time [58]. Usually, the dynamic optimization algorithm is already converged to the current optimal solution when the problem is changed, thus a mechanism is needed to increase the diversity in order to address the convergence issue and search the new optimal solution. Evolutionary computation provides good techniques to solve dynamic optimization problems thanks to their inspiration from biological evolution, which is a continuous adaptation process to dynamic environments and changing conditions of nature.

In the literature of evolutionary dynamic optimization, there does not exist any optimization technique that could solve all types of dynamic optimization problems, which complies with the no free lunch theorems for optimization [59]. Therefore, different EDO techniques have been proposed by researchers in order to solve a broad range of DOPs. Each technique has its own strengths and weaknesses, so each technique suits different types of problems. These techniques are reviewed and categorized in many surveys [6, 57, 58, 60] as follows.

3.5.1. Change Detection in EDO

Many EDO techniques require a mechanism to detect the changes of DOPs in order to be able to search the new optimal solution by applying responsive actions to the change in the environment. This section covers the change detection mechanisms applied in different EDO techniques.

3.5.1.1. Change detection by detectors is a common approach in EDO algorithms that the change of the problem is detected by using some specific individuals as detectors. The EDO algorithm evaluates these detector individuals at each generation, thus, when the fitness values of the detectors are changed, it could be concluded that the problem is changed. These detectors can either be a member of the search population or be independent from the search population. In the former case, the current best solutions or some subpopulations

are preferred as detectors. In the latter case, some fixed points, random solutions, or solutions based on a distribution could be preferred as detectors.

There is a trade-off between the evaluation cost of the detectors and the successful change detection. Mostly, using only one detector provides the detection of change while avoiding the effect of evaluation cost. If the change occurs in only some parts of the problem, increasing the number of detectors might be necessary to a successful detection. On the other hand, it also increases the additional cost of evaluation at every generation regardless of the search of the optimal solution. Thus, it should be well studied by researchers while proposing an EDO algorithm using the change detection by detectors technique.

3.5.1.2. Change detection by algorithm is another approach in EDO algorithms that the change of the problem is detected by monitoring some metrics of the algorithm such as a drop in the average fitness value of the best individuals over generations. This approach does not require any specific evaluation to detect the change, but it might not always detect changes or might cause a false positive when there does not a change actually.

3.5.2. Diversity Introducing Based EDO

EDO algorithms must track the moving global optimum to be able to search the solution after the problem changes. When the algorithm is converged to the current solution, it might cause a negative impact on tracking the next solution because the population might not have any individuals in the area of the moving global optimum. A common approach to this problem is introducing diversity to the population after change detection. It could be achieved via different ways, such as changing the mutation operator by either increasing the mutation rate or the mutation size, changing the crossover operator by increasing the probability, changing the elitism operator by decreasing it, and introducing new random individuals to the population. Multiple methods could be used together to achieve more diversity in the population.

Algorithms using this technique fully focus on searching the current global optimum while the problem is stationary and only deal with the problem changes whenever the changes are detected, which is clearly an advantage. On the other hand, this technique assumes that the change could be detected by the EDO algorithm easily. Identifying the optimal change of evolutionary operators is the main difficulty in this technique because too small changes search only a local area around the current solution and too large changes cause a totally random search in the environment.

3.5.3. Diversity Maintaining Based EDO

Another approach to solve the negative impact of convergence on tracking the moving global optimum is maintaining diversity of the population at a certain level in all generations. This approach does not require to detect changes in the environment explicitly, instead it uses diversified individuals to track and search the new solution. The diversity level of the population could be maintained via different ways, such as adding newly generated random individuals to the population in every generation, placing several specifically distributed sentinel individuals in the entire search space, and selecting individuals based on their distance in addition to their fitness.

Algorithms using this technique are good for solving problems with large and severe changes, because they could start to converge to the area of new global optimum using their diversified individuals. On the other hand, if changes in the problem are small, where the global optimum slightly moves to a nearby place of the previous global optimum, this technique could not be effective. In addition, this technique slows down the convergence while the problem is stationary because of focusing on the diversity of population.

3.5.4. Memory Based EDO

Memory based EDO algorithms store old good solutions in some memory components in order to reuse these old solutions when the DOP is periodically changed by returning to

a nearby place of a previous global optimum. Using a memory of previously found good solutions reduces the convergence time of the algorithm by biasing the population towards the global optimum in the entire search space. In this approach, four aspects of the memory component should be carefully decided, which are the content, the update method, the update period, and the usage of the memory. Each one depends on the type, size, and period of change of the DOP. Algorithms using this technique might not be useful when changes in the environment are not recurrent.

3.5.5. Prediction and Self-adaptation Based EDO

3.5.5.1. Prediction based EDO algorithms try to learn the patterns of previous changes in DOPs and predict the next change of the problem from these patterns. This approach stores a learning model like the memory components of memory-based EDO approach, but it could also work well with the changes that are not cyclic but are predictable. The learning and prediction model varies across algorithms, such as the location of the next global optimum, the locations that new individuals should be introduced to, and the time of the next change.

Algorithms using this technique could be effective in tracking and finding the global optimum quickly when the learning model predicts the change correctly. On the other hand, they might not perform successfully if the DOP changes randomly in the environment or the training data do not represent future changes. In addition, they require a certain amount of time to collect the training data before starting the prediction.

3.5.5.2. Self-adaptation based EDO algorithms try to evolve and adapt the parameters of EA using a learning process like prediction-based algorithms, such as selection ratio, mutation probability, mutation rate, mutation size, crossover probability, crossover rate, and elitism ratio. Adapting parameters during the search process works well when the velocity of change is constant. Otherwise, this approach could not adapt itself quickly to fast changes.

3.5.6. Multipopulation Based EDO

Multipopulation based EDO approaches separate the tasks of exploration and exploitation by assigning the search of the current global optimum to some subpopulations and the track of the changes to others. Algorithms using this technique could maintain the diversity or save a memory by using dedicated subpopulations in order to become more effective for solving DOPs. Adaptively adjusting the number and size of subpopulations, assigning different tasks to subpopulations, and deciding which area of the search space each subpopulation will work on are the main difficulties of this approach [61].

4. 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 4.1.. Section 4.2. presents the proposed solutions based on evolutionary computation techniques for solving problems in ad hoc networks. The use of evolutionary dynamic optimization algorithms in the literature is summarized in Section 4.3..

4.1. Trust Management Systems in VANETs

Researches on trust management systems in VANETs are generally grouped into three categories based on their trust models: they are entity-oriented trust model, data-oriented trust model, and hybrid trust model. In addition, surveys also review the challenges and state the open issues in vehicular networks for trust management solutions. Recently, researchers begin to adapt the machine learning techniques into the trust management methodologies, especially in VANETs.

In a survey [62], the design challenges of cyber security models for V2X communications and the existing security solutions in vehicular networks are presented. The authors stated attacks that target the availability of information are the most dangerous because of causing serious effects on safety-critical situations, while analyzing threats to V2X enabling technologies. The most common trust management methodologies are presented in behaviour-based security solutions category and they are classified into three categories. Assigning different weights for each trust component to evaluate the trust value is the most common trust management category, which is **the weighted-sum method**. This method is considered a lightweight security model, so it is a recommended method for vehicular networks even if setting the weights and trust threshold is challenging for the method. Rewarding cooperative nodes to encourage non-cooperative nodes to behave normally and participate in the network is the second category, **the rewarding-based method**. The main drawback of this method is it can only address the selfish behaviour attack. The third

category is **the fuzzy logic method** where a fuzzy engine applies fuzzy if-then rules on the input variables to evaluate the results according to fuzzy sets and criteria. This method is suitable for predictable vehicular network environments. The authors of the survey mentioned that the use of central units to build centralized security measurement models could not be applied in vehicular networks by taking into consideration that the central units could not cover the whole network and could not always available in distributed networks, although it is a frequently used method in their reviewed security methods. They stated an open issue for trust-based solutions that they should take into consideration the trust attacks where the malicious node behaves intelligently by switching between normal and malicious behaviour to not be detected. Additionally, modeling the real-world traffic environment in the simulation model is still an open issue to attain a high security level in vehicular network communications because the existing security solutions do not take into account the parameters of real environments, so there exists a lack of simulations that apply models of the real-world traffic environment. These trust-related open issues are addressed in the proposed EDO based dynamic trust management model and the real-world application case study section of the experiments in this thesis by implementing an intelligent attacker behaviour in which the malicious vehicles give benign feedbacks as if they are normal vehicles.

Malhi et al. [63] reviewed security challenges, attacks, trust management schemes, and open issues in VANETs in their survey. Most of the security challenges that they stated are related to building a central infrastructure for VANET communication. Using central trusted authorities and short-ranged RSUs has a huge negative impact on the scalability and cost of the VANETs with the difficulty of processing huge amount of data by a central authority. A decentralized approach could be more suitable than building an infrastructure for the implementation of security schemes for VANETs regarding these challenges. The authors pointed out that the high mobility of the vehicles, the high scalability of the VANETs, and the decentralized nature of VANETs are the major challenges for modelling the trustworthiness of vehicles and later classified the existing trust management models into three categories, which are entity-oriented trust models, data-oriented trust models, and hybrid trust models. They compared the properties of existing trust management models

according to the parameters which are decentralized scheme, robustness, authentication, privacy, security, confidentiality, scalability, and dynamicity. It could be shown from the analysis of the trust models in the survey that decentralized scheme, authentication, and confidentiality are the most preferred properties of VANETs to be achieved while proposing a secure model and capturing the dynamicity of the network. None of the analyzed trust models has infrastructure support. The proposed EDO based dynamic trust management model in this thesis targets the decentralized scheme, robustness, security, confidentiality, scalability, and dynamicity properties without the use of a fixed or central infrastructure while proposing a hybrid trust management scheme. The major open issues about the security of VANETs are presented in the survey as detecting the vehicles which switch from benign to malicious while having high trust value and deciding a reasonable reaction time for vehicles without compromising security of the network. Finding the criteria to separate vehicles as trusted or not, achieving a reliable trust calculation, taking actions according to the calculated trust value, and defining the punishment factors are also open problems for evaluating the trust of vehicles. These trust management and security related open issues of VANETs are addressed in the proposed EDO based dynamic trust management model in this thesis.

Hussain et al. [64] reviewed some of the recently proposed trust establishment and management mechanisms in vehicular networks and identified open challenges and research directions for building reliable trust management in VANETs. They first stated that the decision making process of the data shared across the network should be implemented at the node level in decentralized networks, so every node must take part in the trust computation in VANETs. They later discussed the rationale for both the trustworthiness of the vehicle and the trustworthiness of the data and decided that both trust values should be taken into consideration to make any decision based on the information in VANETs. They categorized trust evidences as either direct communication, which is based on its own experience of a node, or indirect communication, which includes collecting recommendations from neighbours and then classified the recent trust management methodologies based on their approaches. It could be seen from the classification that several different techniques are used in trust management methodologies such as fuzzy logic, game theory, blockchain, machine

learning, and others, however, evolutionary computation is not any one of them. Avoiding uncooperative nodes is still an open issue for trust management solutions in VANETs.

Chen and Wei [65] 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 [66]. 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.

Li and Song [67] proposed an attack-resistant trust management scheme for securing vehicular ad hoc networks. They calculate both the data trust and node trust, but they do not build a hybrid trust model based on these. The trustworthiness of a node is defined as a multidimensional vector, but only two trust evidences are used in the calculation of the node trust, which are functional trust and recommendation trust. In addition, the recommendation trust values of each node are aggregated to find the similarities between nodes, which requires a central computation unit.

Yao et al. [68] 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.

Sun et al. [69] proposed a data trust framework for VANETs to detect false data by computing only the truthfulness of messages. They track movements of the vehicle to detect whether the vehicle acts in accordance with its message. Hence, the framework requires vehicles to broadcast their position, velocity, and acceleration data. In addition, the messages sent by the vehicles contain only the claimed acceleration vectors of them in two directions. Besides the lack of entity-based trust management, their proposed method is suitable for limited VANET environments, which is highway scenarios without sharp curves or turns.

In [70], the authors proposed a fuzzy trust model to secure the vehicular network. The values of the three trust evidences are transformed into three fuzzy sets which are named low, medium, and high. The fuzzification is executed using predefined membership functions and the trust level of the message is determined by a fuzzy inference engine which classifies the trust level either acceptable or not acceptable.

In [71], the authors proposed a hierarchical trust management system which involves trust evaluation, trust propagation, and trust aggregation steps. Trust calculation starts with the evaluation of local trust values on each node. The local trust value is determined by only two evidences, the packet forwarding ratio and the delivery ratio. Nodes propagate these local trust values to other nodes that are in the same cluster with them to select a cluster head. All trust opinions of cluster heads are then aggregated to compute the global trust value. They try to cope with the dynamicity of the network by adjusting the weights of the parameters used in calculations.

Zhang et al. [72] proposed an anti-attack trust management scheme in VANET to evaluate the trustworthiness of vehicles under malicious attacks. Vehicles calculate the local trust values by observing whether their neighbours obey the maximum speed rule or not and send the values to a central trusted authority. Most of the work is done by the trusted authority, because it calculates the global trust value of vehicles by using local trust values, social factors, and old global trust values. All values are kept by the central authority and are broadcasted to vehicles on the network. The data trust is not available in this trust management scheme because vehicles do not send any messages to other vehicles.

Machine learning-based trust management approaches for securing the communication of vehicles in vehicular networks have been emerging recently [64, 73]. 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 [74]. 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 [75]. A trust-based deep reinforcement learning (Deep RL) algorithm is proposed to select the most trusted routing path for the communication of connected vehicles [76]. In addition, blockchain based trusted communication systems are proposed to ensure the trustworthiness of vehicles and messages in VANETs [77]. To build a secure intelligent transportation system against unauthorized drivers, another study analyzes and processes drivers' behaviour using deep learning techniques, presenting a different perspective on the problem [78].

An outline of all reviewed trust management systems that focus on decentralized trust models in VANETs is given in Table 4.1. This shows the trust model of each system and the limitation of it. To sum up, the previous studies that focus on developing decentralized trust models in VANETs. They either take into account very limited trust evidences or do not attach much importance to hybrid trust models as shown in Table 4.1. In our initial experiments of this thesis, we proposed a genetic programming based trust management model for VANETs in order to properly evaluate the trustworthiness of data about events [79]. In this thesis, 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.

Table 4.1 Summary of the Related Works about Trust in VANETs domain

Work	Focus	Model	Limitations
[65]	Trust Management	Hybrid Trust	Limited size of trust evidences
[67]	Attack-Resistant Trust Management	Data & Node Trust	Lack of hybrid trust
[68]	Trust Management	Entity & Data Trust	Lack of hybrid trust
[69]	False Message Detection	Data Trust	Lack of entity trust
[70]	Data Securing	Fuzzy Logic	Lack of entity trust
[71]	Trust Management	Entity Trust	Lack of data trust
[72]	Trust Management	Entity Trust	Lack of data trust
[74]	Trust-Aware Intrusion Detection	SVM	Lack of data trust
[75]	Trust-Aware Clustering	K-means	Lack of hybrid trust
[76]	Trusted Routing	Deep RL	Lack of data trust
[77]	Trusted Communication	Blockchain	Lack of hybrid trust
[78]	Driver Identification	DL	N/A
[79]	Trust Management	GP	Not suitable in dynamicity
[80]	False Position Information Detection	ML-based Entity Trust	Lack of data trust & decentralized approach
This Study	Dynamic Trust Management	EDO by GP	Defining change rate of operators

Entity trust values of vehicles are calculated based on the data trust values of messages sent by these vehicles. This research takes into account a broader set of effective trust evidences differently from current trust management researches in VANETs. Moreover, it approaches the problem as a DOP and hence employs the EDO technique to solve it.

4.2. Evolutionary Computation Techniques in Ad Hoc Networks

Many applications of evolutionary computation techniques and other nature-inspired algorithms are employed for different problems in several ad hoc networks. Researchers generally focus on applying these techniques to routing problems in VANETs and there exists

little research on trust management problems in ad hoc networks. Evolutionary computation techniques have not been applied to the trust management problem in VANETs so far.

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 [81]. 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.

In [80], the authors proposed a machine learning based approach to detect position falsification attacks in VANETs. Although they stated that the safety messages are the most important ones and guaranteeing the trustworthiness of the data is the primary concern, they concerned only classifying the vehicles which send false position information as attackers. They use three different combinations of four trust evidences, which are position, speed, difference of position between sender and receiver, and difference of speed between sender and receiver. They define a service plane outside of the vehicles which provides traffic-related services to them in the system model and simulate two machine learning models using an open source dataset that is based on a traffic scenario, thus it can be said that the proposed approach is run in the centralized and offline with global knowledge category.

A survey reviews the applications of evolutionary algorithms that are proposed to solve

optimization problems in MANETs in the literature [82]. 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. Recent researches found in the literature are mostly adopting EC techniques to routing problems. Krundyshev et al. [83] proposed a swarm algorithm to protect the VANET from black hole and wormhole routing attacks. In [84], the authors proposed a GA based routing protocol to find the optimal route path between the source node and the destination node in VANET. Rajeswari et al. [85] proposed a secure routing algorithm using trust-based node selection and a fuzzy inference system in MANETs. Even, there exists a recent survey paper classifying only the nature-inspired optimization algorithms used to solve VANET-related routing problems [86]. Other researches include optimizing data dissemination in DTNs by a multiobjective genetic algorithm [87], optimizing the performance of VANETs by increasing the throughput and decreasing the packet dropping rate using three ANN algorithms [88], and optimizing the trade-off between the performance metric variables in vehicular DTNs using multi-objective PSO [89].

Another survey focuses on the applications of evolutionary computation methods for cybersecurity in MANETs and covers EA, SI, AIS, and EG [90]. 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, and it shows that most of the proposals in the literature are based on EG. Evolutionary computation techniques are investigated for intrusion detection in many studies both for wired and wireless networks [91], such as a genetic programming and grammatical evolution study [92], and a recent SI study which is a grasshopper optimization algorithm [93]. A recent survey reviews the swarm and evolutionary algorithms proposed for intrusion detection systems [94]. Other researches on the applications of EA in ad hoc networks include detecting and removing unhealthy nodes in wireless sensor networks (WSN) using the artificial bee colony algorithm [95].

There exist some studies on the applications of evolutionary algorithms to trust and reputation

systems in the literature, which are proposed mostly for peer-to-peer (P2P) networks. Tahta et al. [96] proposed a trust model evolved by using genetic programming in which a peer calculates the trustworthiness of another peer. In [97], the authors proposed a self-organizing trust model which is evolved using a decimal coded genetic algorithm for peer-to-peer systems. Yuan and Guan [98] proposed an optimized trust-aware recommender system model by using the genetic algorithm to improve the recommending efficiency of existing models. Singh et al. [99] proposed a trust-based intelligent routing algorithm which uses ANN to calculate trust values and enhances the routing performance in DTN. A trust management model is proposed in [100] to recognize trusted nodes in the mobile grid network (MGN) system by using an elitist multiobjective optimization algorithm based on genetic algorithm.

Thakur and Kumar [101] analyzed nature-inspired techniques, classified them according to their source of inspiration, and reviewed them based on their employment in intrusion detection. Genetic algorithm and particle swarm optimization are the most used evolutionary computation techniques in the field of intrusion detection. Other intrusion detection studies have also used ACO, AIS, ABC, and other newly developed nature-inspired algorithms. The analysis of researches shows that nature-inspired techniques reach high detection rates and low false positive rates while offering more flexibility in intrusion detection systems compared to the traditional methods. The authors also list the most significant fields different from intrusion detection, wherein the applications of nature-inspired techniques are developed such as classification, vehicle routing, and clustering, however, they do not include trust and reputation systems.

Sharma and Kaushik [102] presented categories of nature-inspired algorithms and explored the applicability of them in different aspects of the Internet of Vehicles (IoV) network such as vehicle routing, security, and parking space management. Mostly, ACO, PSO, and GA based algorithms are proposed to optimize the routing protocols, improve message transmission, prevent attacks, and find parking slots in VANETs. The analysis of the authors shows that the performance of IoV networks could be optimized by nature-inspired algorithms. They also discussed the open issues and challenges of nature-inspired algorithms in IoV. In order to achieve the optimal performance of a nature-inspired algorithm in an IoV network,

determining the randomness to balance the exploration and exploitation components of the algorithm should be addressed, which is still a challenging task. Other challenging issues include the selection mechanism of one optimal solution among many solutions that have similar fitness values, the tuning and controlling of parameters, choosing the appropriate benchmark functions, and the scalability of the algorithm regarding the high node density in the IoV network.

Mchergui et al. [73] reviewed the strengths and weaknesses of artificial intelligence (AI) techniques which are explored by researches to propose AI-based approaches for VANETs and discussed the open research opportunities of AI techniques in VANETs. They only presented classical machine learning (ML), deep learning (DL), and SI as AI techniques, hence EC techniques are not mentioned in this taxonomy. Some of the various problem areas of VANET where the AI techniques could be applied to are identified as routing decision/optimization, traffic signal management, misbehaviour detection, attack detection/prevention, clustering, driver behaviour prediction, intrusion detection, traffic congestion prediction/handling, accident prediction, and malicious node detection. ML algorithms are stated as appropriate tools in order to develop trust models in the area of trust management in VANETs, considering their advantage on various non-linear classification scenarios and the massive data that are generated in vehicular environment. The highly dynamic network topology and the large scale of the network are stated as major characteristics of the VANETs that should be taken into consideration while applying SI techniques to problems in VANETs, which is also applicable for EC techniques.

To sum up, the current research is the first application of evolutionary computation techniques to the trust management problem in ad hoc networks, as far as we know. 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 evolutionary computation techniques to choose the parameters automatically from a broader set and change them according to the dynamicity. Evolutionary computation algorithms require fewer a priori assumptions about the problem at hand [10]. Furthermore, evolutionary computation seamlessly lends itself to the integration of human expert knowledge as needed, and the representation of solutions in evolutionary computation algorithms can be quite flexible [10]. These characteristics of evolutionary computation are among the main motivations behind using evolutionary computation in this research.

4.3. 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 [6]. 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 the techniques of change detection, diversity introducing, diversity maintaining, memory usage, prediction, self-adaptation, and multipopulation. It is also discussed that little attention has been given to the application of EDO for solving real-world DOPs and there exist a limited number of studies focused on real-world applications of EDO [6, 103]. Most of the existing EDO studies on real-world applications are either using GA for DOPs of different areas in MANETs or ACO for DOPs of areas other than ad hoc networks.

Yang et al. [104] state that the dynamic topology change of the network over time is one of the most important characteristics in MANETs because of either the mobility or the energy conservation of the nodes. This brings out that the general static shortest path routing problem in MANETs becomes a dynamic optimization problem, so they propose to use the diversity maintaining and memory usage techniques of EDO with GAs to solve the DOP. They show that this EDO algorithm can quickly adapt to the changes of the network topology and produce good solutions dynamically after the changes. Cheng and Yang [105] propose

another approach of the EDO with GAs to solve the dynamic shortest path routing problem in MANETs. It is shown that the multipopulation and diversity maintaining techniques of EDO could provide good solutions after the changes in the environment while adapting to the changing network topology dynamically and quickly. Cheng and Yang [106] also propose to use the EDO with GAs to solve the dynamic multicast problem in MANETs where the network topology is changed because nodes either enable or disable themselves with the concern of energy conservation. They only use the diversity maintaining technique and show that it can also adapt to changes quickly and find good solutions after each change.

Cheng et al. [107] describe the dynamic load balanced clustering problem in MANETs as a DOP considering the dynamic network topology changes due to the movement of nodes or energy conservation. Then, they propose to use several dynamic GAs which include either only one or a combination of the diversity maintaining, memory usage, and multipopulation techniques of EDO, to solve the DOP. The results of these dynamic GAs show that these EDO techniques work well in the dynamic real-world MANETs. Cheng and Yang [108] state that both the shortest path routing problem and the multicast routing problem become real-world DOPs in MANETs because of one of the most important characteristics of MANETs, which is the dynamically changing network topology due to the mobility of nodes or the energy conservation. They investigate several dynamic GAs to solve the dynamic routing problems in MANETs by integrating different EDO techniques into the GA, which are diversity maintaining, memory usage, and combination of them. These EDO techniques make the dynamic GAs are able to quickly adapt to changes in the dynamic environment and provide good solutions after the environment is changed, which is shown by the experimental results.

Chitty and Hernandez [109] introduce a hybrid dynamic ACO algorithm to solve the dynamic vehicle routing problem. They give the reason that makes the problem of routing a vehicle from a starting point to a destination point through a road network is a DOP, which is continuously changing road conditions as a result of dynamic events such as traffic congestion or blocked roads. They use the diversity maintaining technique of EDO by storing and updating the pheromone for routing paths from each node in the network to the destination node, without concerning only the route between the start node and the

Table 4.2 Summary of the Related Works in domains apart from Trust in VANETs

Work	**Focus**	**Domain**	**Model**
[83]	Routing	VANETs	Swarm
[84]	Routing	VANETs	GA
[85]	Trusted Routing	MANETs	Fuzzy Logic
[87]	Data Dissemination	DTNs	Multiobjective GA
[88]	Communication Performance	VANETs	ANN
[89]	Performance Optimization of Routing	Vehicular DTNs	Multi-objective PSO
[92]	Intrusion Detection	MANETs	GP & GE
[93]	Intrusion Detection	Network	GOA
[95]	Unhealthy Node Detection	WSN	ABC
[96]	Trust & Reputation	P2P	GP
[97]	Trust-Based Recommendation	P2P	GA
[98]	Trust-Aware Recommendation	Network	GA
[99]	Trust-Based Routing	DTNs	ANN
[100]	Trust Management	MGN	Multiobjective GA
[104]	Routing	MANETs	EDO by GA
[105]	Routing		
[106]	Multicasting		
[107]	Clustering		
[108]	Routing		
[109]	Routing	Vehicles	EDO by ACO
[110]	Routing		
[111]	Dynamic Travelling Salesman	DOPs	
Thesis	Dynamic Trust Management	VANETs	EDO by GP

destination node. This complete solution space helps in finding the new optimal routing path when it is changed. The experimental tests on randomly created networks show that the proposed algorithm outperforms the standard algorithm in terms of finding the optimal route path and allowing instantaneous rerouting after each dynamic change within the road network.

Xing et al. [110] propose a hybrid ACO algorithm to solve the extended capacitated

arc routing problem by adjusting the selection probabilities of parameter combinations dynamically and show that it is more effective than the existing algorithms. Mavrovouniotis and Yang [111] propose an ACO framework using the diversity maintaining and memory usage techniques of EDO to solve the dynamic travelling salesman problem with traffic factors, which is described as a DOP. The experimental results show that the proposed algorithms perform well in dynamically changing environments.

An outline of all reviewed researches that focus on evolutionary computation techniques and evolutionary dynamic optimization algorithms in ad hoc networks is given in Table 4.2. This shows the ad hoc network domain of each research and the algorithm model that preferred in the research. In this thesis, we adapted evolutionary dynamic optimization techniques using genetic programming to the trust management problem in VANETs and proposed a dynamic trust management model for vehicular ad hoc networks. 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.

5. PROPOSED METHOD

This thesis aims to explore the use of evolutionary computation and evolutionary dynamic optimization techniques in order to provide a dynamic trust management framework that automatically generates the trust formula. The purpose of this framework is to prevent bogus message attacks in VANETs while detecting changes in the environment and reacting to them. The network environment used in this study is introduced in Section 5.1.. The proposed dynamic trust management method is given in details in Section 5.2.. Section 5.3. gives the details of the dynamic evolution process of trust formula used in the proposed method.

5.1. 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.

5.1.1. 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 [65]. 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 [65, 68].

5.1.2. Application Messages

Many applications running on VANETs mainly focus on sharing information about events that vehicles come across [112]. 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.

5.1.2.1. 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 5.1.

Table 5.1 Format of the Beacon Message

Unique Identifier	Vehicle Type	Message Time	Current Position	Current Velocity

5.1.2.2. 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 5.2. 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 5.2 Format of the Event Message

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

5.1.3. 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.

5.1.3.1. 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.

5.1.3.2. 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.

5.2. 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.

5.2.1. 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.

5.2.2. Trust Properties

Trust management systems for ad hoc networks should take into account all five properties of trust, dynamicity, context-dependency, subjectivity, asymmetry, and incomplete/partial transitivity, as mentioned in Section 2.6..

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.

5.2.3. 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 5.3 shows the trust evidence set and Table 5.4 shows the notations used in the proposed dynamic trust management model, which is described in detail below.

5.2.3.1. 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

Table 5.3 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

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. 2, the percentage of added neighbours of the vehicle A, AP_A is defined as in Eq. 3, and the percentage of removed neighbours of the vehicle A, RP_A is defined as in Eq. 4:

$$ND_A = NN_A \,/\, MN_A \tag{2}$$

$$AP_A = AN_A \,/\, MN_A \tag{3}$$

$$RP_A = RN_A \,/\, MN_A \tag{4}$$

5.2.3.2. 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. 5, the proximity of receiver vehicle R to the sender vehicle S VP_R^S is defined as in Eq. 6, the proximity of the sender vehicle S to the event X SP_S^X is defined as in Eq. 7 and the proximity of event time X to current time TP_X is defined as in Eq. 8:

$$EP_R^X = (MD \ - \ ED_R^X) \ / \ MD \tag{5}$$

$$VP_R^S = (MD \ - \ VD_R^S) \ / \ MD \tag{6}$$

$$SP_S^X = (MD \ - \ ED_S^X) \ / \ MD \tag{7}$$

$$TP_X = (MT - (T - GT_X)) \ / \ MT \tag{8}$$

5.2.3.3. 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

Table 5.4 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

as in Eq. 9:

$$W_V(x) = \begin{cases} 1.0, & when\ x\ is\ a\ police\ car \\ 0.7, & when\ x\ is\ a\ public\ service\ vehicle \\ 0.5, & when\ x\ is\ an\ ordinary\ automobile \end{cases} \tag{9}$$

5.2.3.4. 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 5.1.2.2.. 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. 10:

$$W_E(x) = \begin{cases} 1.0, & when\ x\ is\ a\ safety\ event \\ 0.8, & when\ x\ is\ an\ efficiency\ event \\ 0.5, & when\ x\ is\ an\ infotainment\ event \end{cases} \tag{10}$$

5.2.3.5. 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. 11:

$$PE_A^X = EN_A^X \ / \ MN_A \tag{11}$$

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. 12:

$$PT_A^X = TN_A^X \ / \ EN_A^X \tag{12}$$

Therefore, the event X in Eq. 11 corresponds to all received event messages at the same position regardless of their event types and the event X in Eq. 12 corresponds to messages that have the same position and the same event type.

5.2.3.6. 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.

5.2.3.7. 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. 13 and the average weight of events at the same location X sent to vehicle A EW_A^X is defined as in Eq. 14:

$$VW_A^X = (\sum_{i=1}^{EN_A^X} W_V^i) \; / \; EN_A^X \tag{13}$$

$$EW_A^X = (\sum_{i=1}^{EN_A^X} W_E^i) \; / \; EN_A^X \tag{14}$$

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. 15 and the average weighted trust value of event X by vehicle R TE_R^X is defined as in Eq. 16 (i means each forwarder vehicle):

$$TV_R^S = (\sum_{i=1}^{EN_A^X} VT_R^i * VT_i^S)/EN_A^X \tag{15}$$

$$TE_R^X = (\sum_{i=1}^{EN_A^X} VT_R^i * ET_i^X)/EN_A^X \tag{16}$$

5.2.3.8. 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 5.2.4.. 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.

5.2.4. 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 neighbour vehicles is stored in vehicles.

5.2.5. 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 5.5 shows the forwarded event message format.

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

Table 5.5 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	

malicious event message, and the vehicle trust value of the attacker in addition to information about the owner of this negative opinion. Table 5.6 shows the format of the negative opinion message.

Table 5.6 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		

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

5.2.6. 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. 17 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.

$$T_R^S = \begin{cases} CT_R^S \times ET_R^S, & 0 \leq ET_R^S < TT \\ CT_R^S + (1 - CT_R^S) \times (\dfrac{ET_R^S - TT}{1 - TT}), & TT \leq ET_R^S \leq 1 \end{cases} \tag{17}$$

While calculating T_R^S, a well-known principle about trust "hard to earn but easy to lose" [26, 28, 29] 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. 17 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. 18. 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.

$$T_R^S = CT_R^S - CT_R^F \times (CT_R^S \times (1 - ET_F^S)) \tag{18}$$

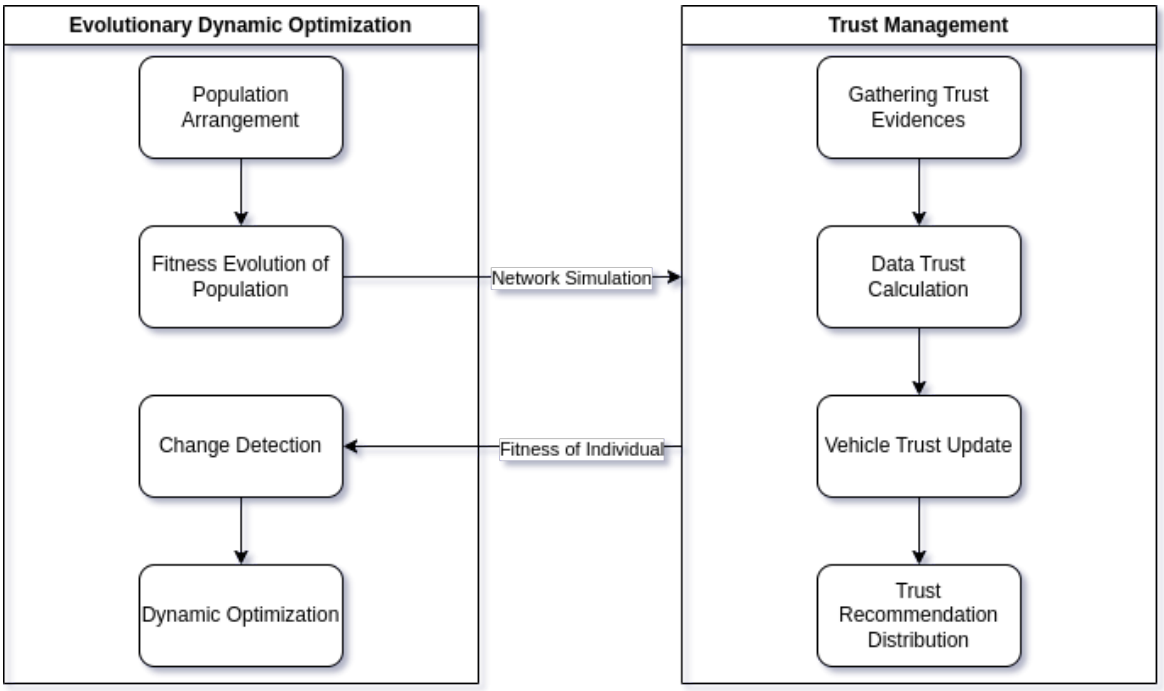


Figure 5.1 The dynamic trust management framework with evolutionary dynamic optimization

5.3. Evolution and Dynamic Optimization of Trust Formula

The trust formula is evolved iteratively by using genetic programming and dynamically optimized according to changes in the environment by using evolutionary dynamic optimization techniques. Operators of the genetic programming provide better trust formulas while the environment is stationary and the problem remains unchanged from the DOP view, which is the exploitation part of the process. When the problem environment is changed, dynamic optimization techniques increase and maintain the diversity of population in order to search the location of new optimum trust formula, which is the exploration part of the process. The pseudocode of the single-objective genetic programming with evolutionary dynamic optimization used in this study is given in Alg. 2. The dynamic trust management framework with evolutionary dynamic optimization is shown in Figure 5.1.

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 other evolutionary computation algorithms in this

Algorithm 2 The pseudocode of the single-objective GP with EDO

```
generate the initial population of individuals randomly
while a termination criterion is not satisfied do
    evaluate the fitness value of each individual in the population
    check for changes in the fitness value of best individual
    if a change is detected in the problem environment then
        increase the diversity of population by changing the parameters of operators
        maintain the diversity of population by adding new diversified individuals
    end if
    select the fittest individuals for reproduction
    breed new individuals through crossover and mutation operators
    replace the least-fit individuals of the population with new individuals
end while
return best-of-run individual as the solution to the problem
```

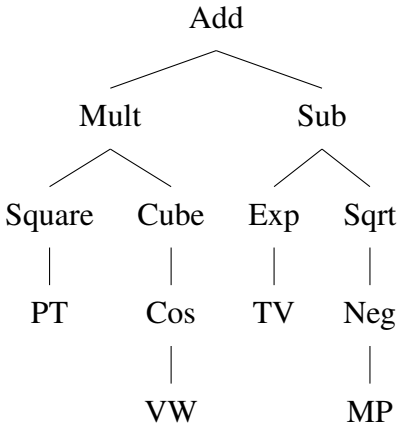


Figure 5.2 The GP tree of a simple trust formula including trust evidences and operations

study. In-order traversal of the tree outputs a candidate formula. Terminal nodes of the tree are trust evidences in Table 5.3 and some ephemeral random constants (ERC). Non-terminal nodes of the tree consist of the mathematical operations listed in Table 5.7. 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 5.2. This tree corresponds to the following formula given in Eq. 19.

$$\{PT^2 \times [(\cos(\pi \times VW) + 1)/2]^3 + [(e^{TV} - 1)/(e - 1) - \sqrt{1 - MP} + 1]/2\}/2 \quad (19)$$

Table 5.7 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	$(\cos(\pi X) + 1) / 2$

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 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) [113], defined as in Eq. 20.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{20}$$

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 [114–117]. 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 [6]. 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. In this study, the dynamic optimization problem is determined as the change in the number of events in the VANET environment from a low density to a high density. Besides the naturally dynamic topology of VANET, the increasing number of events causes more event messages are spread across the network. Attack distribution of malicious vehicles and density of the malicious messages are also changed in the meantime, so this change should be detected and the evolved trust formula

should be dynamically altered according to environmental changes. 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 [118–120]. 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 [6]. 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 [6]. It is shown in [121] that these adaptive control parameters help GP to perform better than the static control parameters in dynamic problems. When a change is detected, the proposed dynamic trust management model in this study applies the commonly used EDO approaches, which are diversity introducing and diversity maintaining, by simply increasing the diversity of population via increasing the mutation rate and introducing new 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.

6. EXPERIMENTAL RESULTS

The proposed evolutionary dynamic optimization based dynamic trust management model in this thesis is tested with both synthetic network simulation scenarios and a real-world traffic model simulation. These experiments and the achieved results are detailed in this chapter. Section 6.1. presents the experimental settings, scenarios regarding attacks, and dynamic changes employed in the network simulations. Section 6.2. presents and discusses the experimentals results. Section 6.3. discusses a case study using the proposed model on a traffic model taken from real world.

6.1. 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.

6.1.1. 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, which is a discrete-event computer network simulator for internet systems [122]. The ECJ toolkit, which is a java-based evolutionary computation research system [123], is used for EC implementation.

Parameter Name	**Parameter 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)

Table 6.1 Network Simulation Parameters

6.1.1.1. Network Properties The parameters used in the network simulations are listed in Table 6.1. The values of parameters other than the application-specific ones are chosen in accordance with a previous trust management model for VANETs [67]. Each candidate trust formula in GP is evaluated on three networks. In each network simulation, the random waypoint mobility model, in which the initial positions of nodes are determined randomly and nodes move to random directions using random velocity and acceleration values [124], 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 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. 21. 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 6.2.

$$O(I \times G \times F) \tag{21}$$

Parameter Name	**Parameter 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

Table 6.2 EDO Parameters

6.1.1.2. EDO Properties Table 6.2 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" [6] and the use of the current best solution as the detector is a common approach [118–120].

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 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" [6]. 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.

6.1.2. 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,

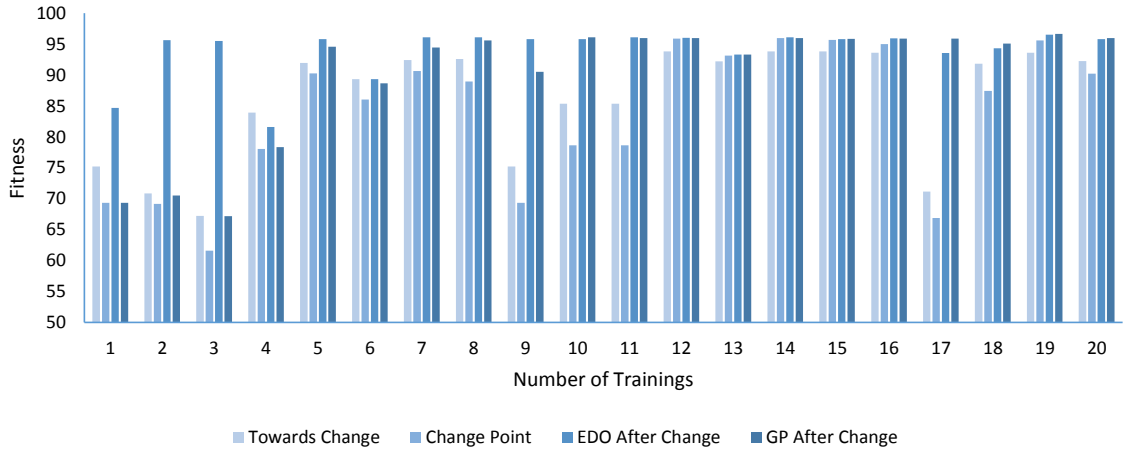


Figure 6.1 Performance of the best individuals before and after the change in the problem

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.

6.2. 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 [125]. 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 6.3.

Table 6.3 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

Table 6.4 Fitness Values of the Best Individuals in Figure 6.1

	Towards Change	Change Point	EDO After Change	GP After Change
1	75.20%	69.32%	84.70%	69.32%
2	70.87%	69.17%	95.63%	70.52%
3	67.21%	61.61%	95.49%	67.18%
4	83.91%	78.06%	81.61%	78.35%
5	91.95%	90.27%	95.81%	94.58%
6	89.35%	86.05%	89.34%	88.64%
7	92.42%	90.64%	96.08%	94.46%
8	92.60%	88.95%	96.08%	95.59%
9	75.20%	69.32%	95.79%	90.51%
10	85.37%	78.62%	95.79%	96.08%
11	85.37%	78.62%	96.08%	95.96%
12	93.83%	95.88%	95.99%	95.98%
13	92.21%	93.12%	93.28%	93.28%
14	93.83%	95.96%	96.07%	95.96%
15	93.83%	95.68%	95.81%	95.83%
16	93.59%	94.99%	95.94%	95.89%
17	71.17%	66.87%	93.55%	95.88%
18	91.80%	87.42%	94.33%	95.08%
19	93.61%	95.58%	96.51%	96.65%
20	92.25%	90.21%	95.81%	95.96%

6.2.1. Performance of the Best Individuals

Figure 6.1 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

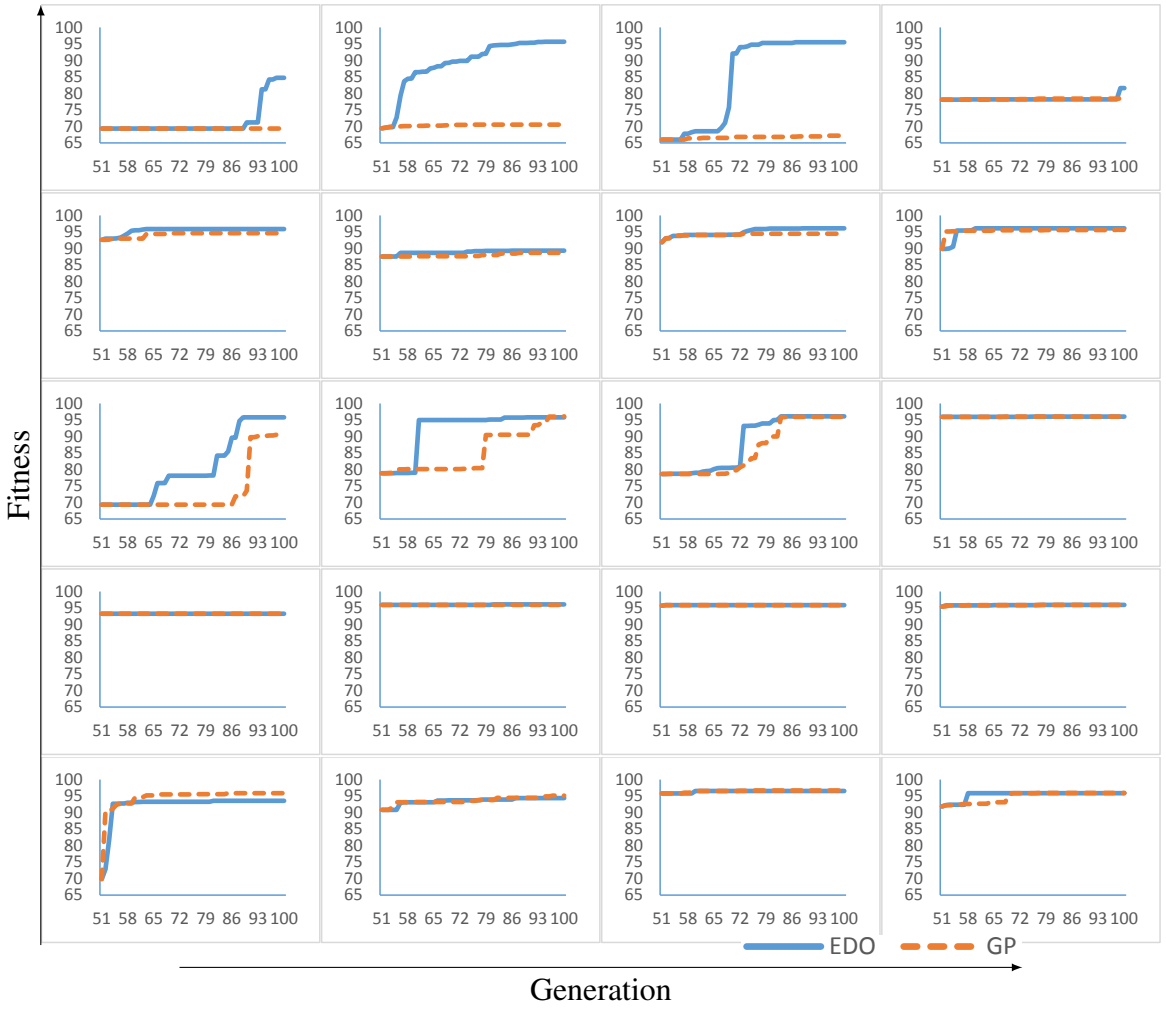


Figure 6.2 Convergence graphs of all training phases after 50^{th} generation for both EDO and GP

GP, respectively. Please note that all evolved individuals are evaluated by using all three networks described in Section 6.1.1.1., so the fitness values in the Figure 6.1 are the average value of the results in the three networks.

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

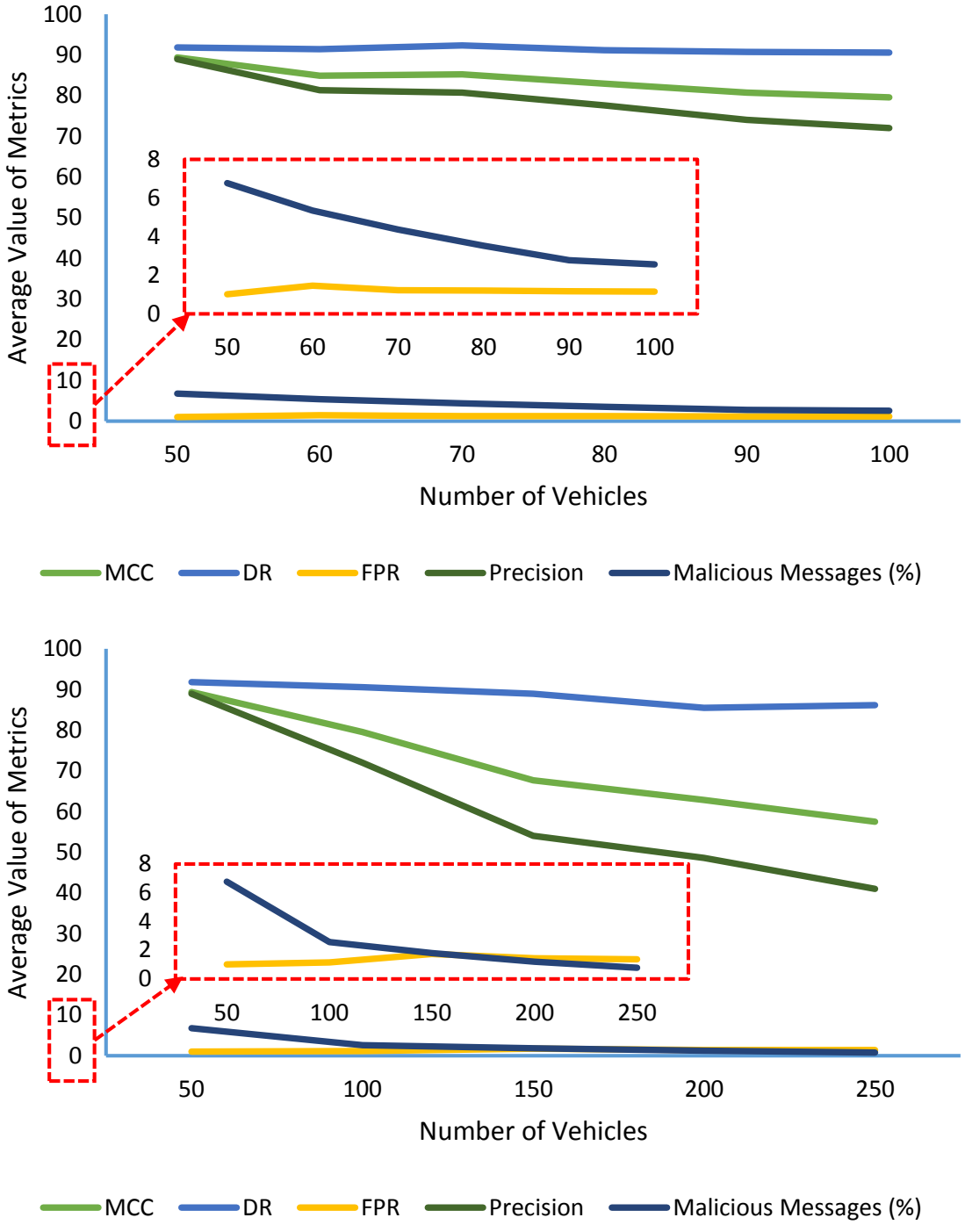


Figure 6.3 Performance of the model on networks with higher density of benign vehicles

or equal individuals compared to GP. Besides, the EDO finds fitter individuals quicker than the GP as shown in the convergence graphs in Figure 6.2, which shows the fitness value of the best individual in each generation after the problem change point.

Table 6.5 Values of the Metrics in Figure 6.3

Vehicle Number	MCC	DR	FPR	Precision	Malicious Message
50	89.38%	91.81%	1.01%	88.95%	6.76%
60	84.89%	91.42%	1.46%	81.29%	5.35%
70	85.26%	92.32%	1.21%	80.76%	4.36%
80	82.89%	91.17%	1.20%	77.62%	3.53%
90	80.74%	90.74%	1.16%	74.04%	2.77%
100	79.56%	90.57%	1.15%	72.02%	2.56%
150	67.65%	88.98%	1.73%	54.02%	1.81%
200	62.82%	85.49%	1.43%	48.58%	1.20%
250	57.51%	86.12%	1.37%	41.00%	0.77%

6.2.2. Performance on Networks with Higher Density of Benign Vehicles

Figure 6.3 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 level afterwards even if the detection rate (DR) (i.e., recall) decreases only 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.

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

Figure 6.4 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 6.3, 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 6.3 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 detection of malicious messages while the total benign event messages in the environment increases.

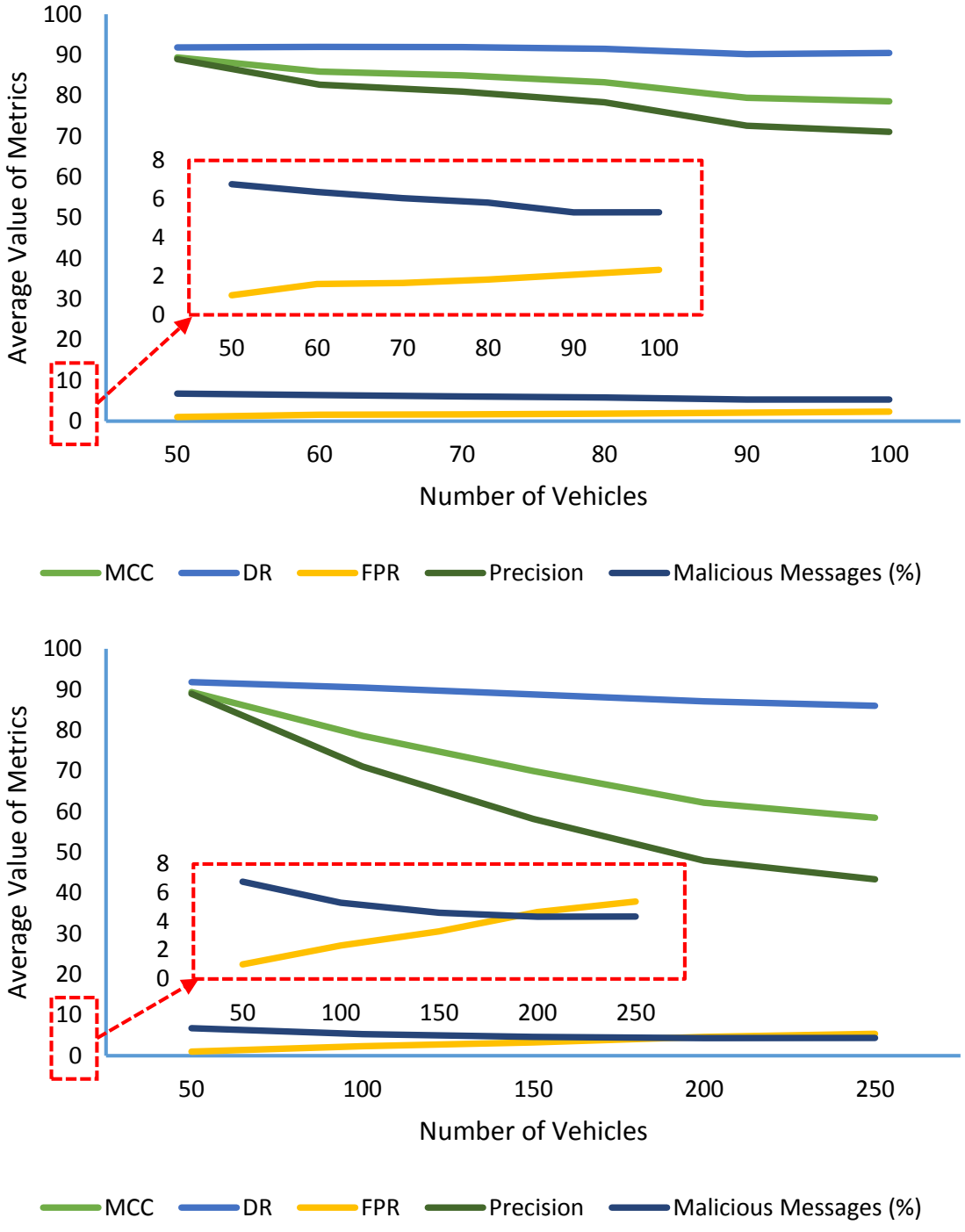


Figure 6.4 Performance of the model on networks with higher density of vehicles & attackers

Table 6.6 Values of the Metrics in Figure 6.4

Vehicle Number	MCC	DR	FPR	Precision	Malicious Message
50	89.38%	91.81%	1.01%	88.95%	6.76%
60	85.87%	91.96%	1.59%	82.70%	6.36%
70	84.93%	91.95%	1.65%	80.95%	6.05%
80	83.27%	91.52%	1.82%	78.33%	5.81%
90	79.43%	90.19%	2.08%	72.61%	5.30%
100	78.59%	90.47%	2.33%	71.09%	5.31%
150	69.93%	88.82%	3.30%	58.17%	4.60%
200	62.20%	87.05%	4.64%	47.95%	4.33%
250	58.48%	85.99%	5.38%	43.35%	4.33%

6.2.4. Performance on Networks with Higher Density of Events

Figure 6.5 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. 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 6.3 and 6.4 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

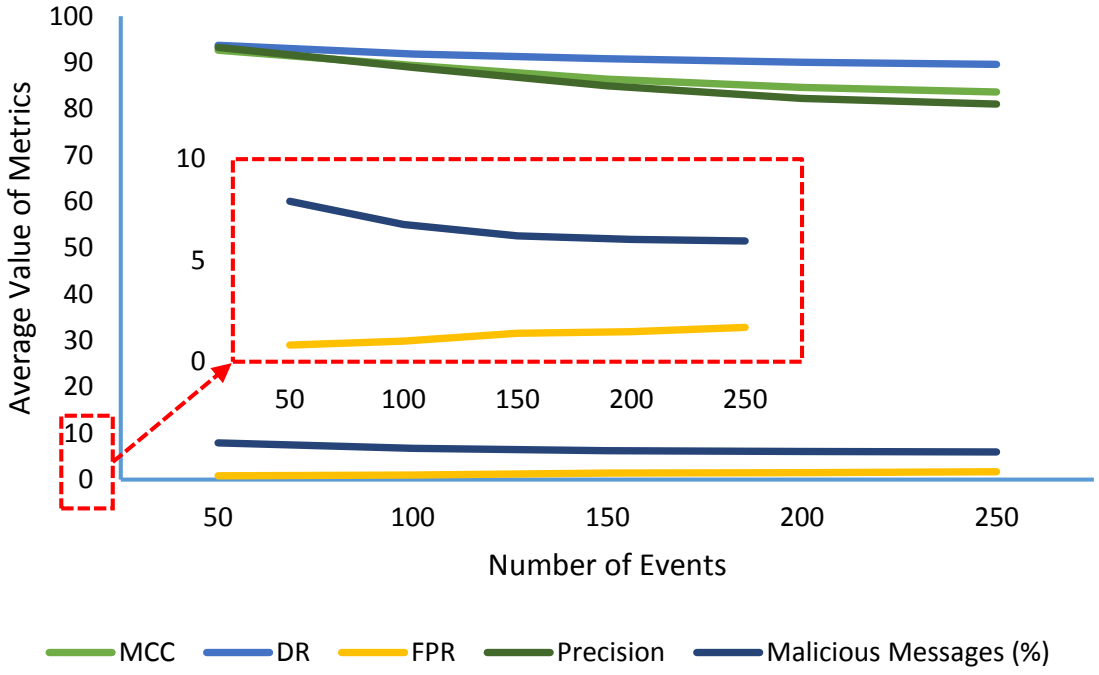


Figure 6.5 Performance of the model on networks with higher density of events

Table 6.7 Values of the Metrics in Figure 6.5

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%

previous scenarios where the density of vehicles/attackers is increased. As we compare Figure 6.5 with Figures 6.3 and 6.4, 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.

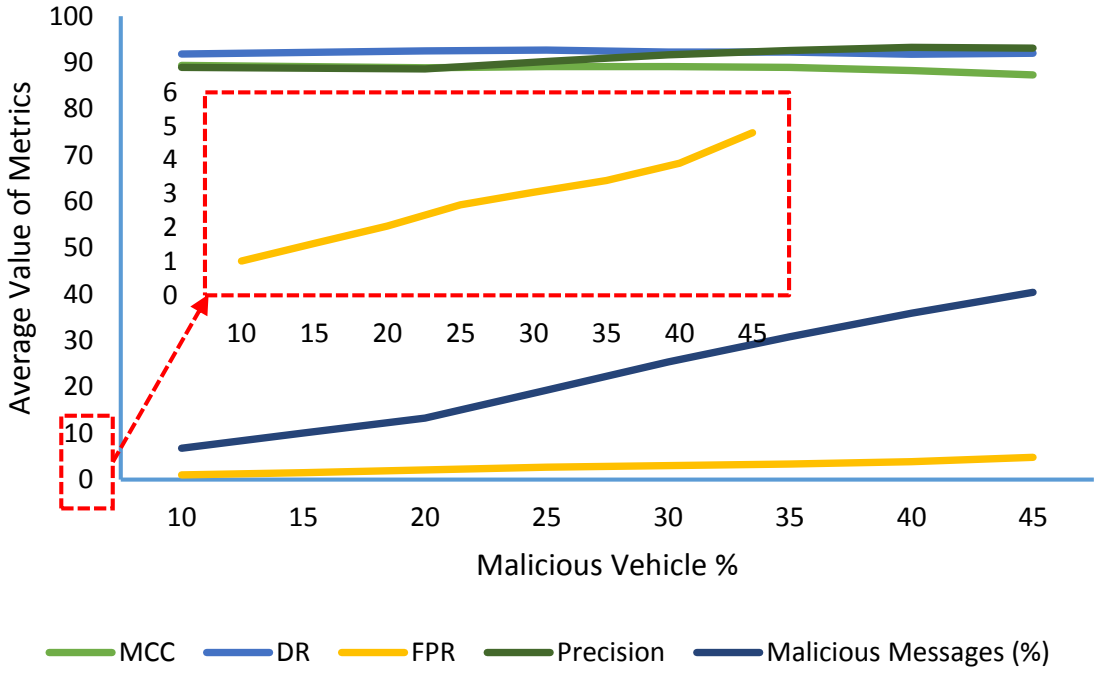


Figure 6.6 Performance of the model on networks having more malicious nodes

6.2.5. Performance on Networks with Higher Density of Attackers

Figure 6.6 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 percentage of FPR only increases from 1.01% to 4.80% because of the decreasing number of benign messages.

Table 6.8 Values of the Metrics in Figure 6.6

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%

As shown in the Figure 6.5 that the fitness value is maintained at a level when the ratio of malicious messages does not decrease much, Figure 6.6 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 an attack-resistant trust management scheme for securing VANETs, named as ART [67], which tries to detect and cope with malicious messages by evaluating the trustworthiness of both data and vehicles and is implemented in a similar attack scenario as stated in Section 6.1.1.1.. When more attacks exist in the network, the precision and recall values of ART decrease as stated in [67]. 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 6.8. 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 misclassifying of a malicious attacker could result in drastic results in traffic.

Table 6.9 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)

6.3. 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 [126] 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 6.1 are listed in Table 6.9.

The fitness value of the best individuals in the training of this experiment in the format of Table 6.4 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 [127], 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.

6.4. 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 [6].

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 [74]. Additionally, it is shown that reinforcement learning (RL) is a promising approach for processing large amounts of data sent from vehicles in VANETs [76]. In the future, the use of deep reinforcement learning techniques on the problem could be explored. Chen et al. [128] 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.

7. CONCLUSION

This chapter summarizes the conducted research, reviews the contributions, and completes the thesis. The scope and motivation of the thesis are revisited to show that the work done in this study supports them. Finally, future work areas are also discussed.

7.1. Summary of the Research

The complexities and difficulties of trust management in VANETs are reviewed and the main problems of existing approaches are outlined in Chapter 4.. The main issues can be summarized as follows:

- The management of trustworthiness of only the nodes in MANETs and other ad hoc networks is no longer suitable for VANETs. The trustworthiness of both vehicles and data in the traffic should be dealt with.
- The detection of attacks is a complex problem in ad hoc networks. Having a more dynamic topology than other ad hoc networks makes malicious vehicle detection harder in VANETs.

Researchers have generally focused on either the entity trust or the data trust. In this thesis, both trust values are taken into consideration by using a hybrid trust model. In addition, researchers generally choose a fixed and very limited set of trust evidences to build a trust management system, but changes in the environment invalidate the chosen trust evidences. The parameters of the trust management model are chosen automatically from a broader set of trust evidences by using evolutionary computation techniques and are changed according to dynamicity in the network by using evolutionary dynamic optimization techniques in order to take into consideration the dynamically changing environment of VANETs.

This thesis 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.

The performance of the trust formula, that is dynamically evolved by GP and EDO, is evaluated on different simulated network scenarios with varying conditions of network topologies, mobility patterns, event positions, and vehicle, event, and attacker densities. The proposed technique shows a good performance for detecting bogus information attacks. The best evolved trust formula achieves 89.38% MCC, 91.81% DR, and 1.01% FPR, which could be seen a good detection of malicious activity in the network traffic. Effects of the increasing density of vehicles, events, and attackers to detection of malicious messages are discussed. The best trust formula obtains 87.33% MCC, 92.01% DR, and 4.8% FPR when $\approx$ 40% of the network traffic is malicious, which demonstrates its robustness to increasing malicious messages.

The performance of proposed EDO based dynamic trust management model on a real-world traffic model simulation is also presented in this study in order to reduce the gap between synthetic environments with real-world applications. The best trust formula obtains 80.68% MCC and 0.14% FPR values in test networks using the real-world traffic data model with different event positions. 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.

7.2. Contributions of the Thesis

The main contributions of this thesis are outlined as follows:

- **Evolutionary computation techniques for trust management in VANETs:** This research investigates the generation of trust calculation formula automatically rather than using a predefined static one. Evolutionary computation techniques evolve the trust calculation formula by evaluating candidate trust formulas and applying genetic operators. This thesis shows that GP could evolve effective trust formulas in order to distinguish bogus information from legitimate messages and attackers from benign vehicles.

- **Effectiveness of trust evidences in trust management systems:** This research presents the use of a broader set of trust evidences. It aims to satisfy the requirements of trust management systems in VANETs and to well-represent the network.

- **Evolutionary dynamic optimization techniques for dynamic trust management model:** This research proposes a novel approach by discovering different evolutionary dynamic optimization techniques. Our main contribution in this thesis is to detect the dynamically changing environment in the VANET network and react to the changes automatically by dynamically adapting the evolved trust formula.

- **Opportunities of real-world applications in VANETs:** This thesis investigates running the proposed model on a real-world traffic model in addition to synthetic test networks. This provides a bridge to reduce the gap between the theoretical EDO research and applications of real-world DOPs.

The work presented in this thesis is the first study that investigates the use of EC techniques for trust management in VANETs, defines the trust management problem from the dynamic optimization problem point of view, and solves the DOP in VANETs using EDO techniques to the best of our knowledge.

7.3. Future Research

The potential areas for future research are summarized below:

- **Applying different machine learning techniques to trust management:** In this research, we show applying evolutionary computation techniques to the trust management problem. Different machine learning techniques could be used to build trust management models such as reinforcement learning [129]. Deep reinforcement learning and transfer learning could be investigated in order to adapt the model to a new, more dynamic environment in the future.

- **Exploration of new attacks:** More research is needed to implement more complex attack scenarios such as on-and-off and collaborative attacks. The techniques proposed in this thesis could be used to evolve different trust calculation formulas in order to detect a variety of new attacks. Malicious vehicles cease executing their attacks for a short time during the on-and-off attack scenario. They become trusted by other vehicles in the network by behaving benignly in that period. In collaborative attacks, malicious vehicles might support one another by sending malicious messages of attackers to other vehicles. Additionally, malicious vehicles could perform attacks to the distribution of vehicle trust value by sending unrealistic negative opinions about benign vehicles. In the future, further investigation of our proposed approach is needed in order to detect the malicious vehicles in such cases, which becomes more challenging.

To conclude, we believe that evolutionary dynamic optimization based evolutionary computation approaches to develop a dynamic trust management model are of significant potential benefit for the detection of malicious activities and attacks in dynamically changing environments such as VANETs, and we encourage the research community to explore their use.

REFERENCES

[1] Georgios Karagiannis, Onur Altintas, Eylem Ekici, Geert Heijenk, Boangoat Jarupan, Kenneth Lin, and Timothy Weil. Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions. *IEEE communications surveys & tutorials*, 13(4):584–616, **2011**. doi:10.1109/SURV.2011.061411.00019.

[2] Saif Al-Sultan, Moath M Al-Doori, Ali H Al-Bayatti, and Hussien Zedan. A comprehensive survey on vehicular ad hoc network. *Journal of network and computer applications*, 37:380–392, **2014**. doi:10.1016/j.jnca.2013.02.036.

[3] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Networks*, 61:33–50, **2017**. doi:10.1016/j.adhoc.2017.03.006.

[4] Xinxin Fan, Ling Liu, Rui Zhang, Quanliang Jing, and Jingping Bi. Decentralized trust management: Risk analysis and trust aggregation. *ACM Computing Surveys (CSUR)*, 53(1):1–33, **2020**. doi:10.1145/3362168.

[5] Ayyoub Lamssaggad, Nabil Benamar, Abdelhakim Senhaji Hafid, and Mounira Msahli. A survey on the current security landscape of intelligent transportation systems. *IEEE Access*, 9:9180–9208, **2021**. doi:10.1109/ACCESS.2021.3050038.

[6] Trung Thanh Nguyen, Shengxiang Yang, and Juergen Branke. Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24, **2012**. doi:10.1016/j.swevo.2012.05.001.

[7] Elvin Eziama, Kemal Tepe, Ali Balador, Kenneth Sorle Nwizege, and Luz MS Jaimes. Malicious node detection in vehicular ad-hoc network using machine learning and deep learning. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, **2018**. doi:10.1109/GLOCOMW.2018.8644127.

[8] Sarah Ali Siddiqui, Adnan Mahmood, Quan Z Sheng, Hajime Suzuki, and Wei Ni. A survey of trust management in the internet of vehicles. *Electronics*, 10(18):2223, **2021**. doi:10.3390/electronics10182223.

[9] Bernabé Dorronsoro, Patricia Ruiz, Grégoire Danoy, Yoann Pigné, and Pascal Bouvry. *Evolutionary algorithms for mobile ad hoc networks*. John Wiley & Sons, **2014**. doi:10.1002/9781118833209.

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

[11] Amal Hbaieb, Samiha Ayed, and Lamia Chaari. A survey of trust management in the internet of vehicles. *Computer Networks*, 203:108558, **2022**. doi:10.1016/j.comnet.2021.108558.

[12] Shrikant S Tangade and Sunilkumar S Manvi. A survey on attacks, security and trust management solutions in VANETs. In *2013 Fourth international conference on computing, communications and networking technologies (ICCCNT)*, pages 1–6. IEEE, **2013**. doi:10.1109/ICCCNT.2013.6726668.

[13] Hannah Lim Jing Ting, Xin Kang, Tieyan Li, Haiguang Wang, and Cheng-Kang Chu. On the trust and trust modeling for the future fully-connected digital world: A comprehensive study. *IEEE Access*, 9:106743–106783, **2021**. doi:10.1109/ACCESS.2021.3100767.

[14] Zhiquan Liu, Jianfeng Ma, Zhongyuan Jiang, Hui Zhu, and Yinbin Miao. LSOT: A lightweight self-organized trust model in VANETs. *Mobile Information Systems*, 2016, **2016**. doi:10.1155/2016/7628231.

[15] Muhammad Mohsin Mehdi, Imran Raza, and Syed Asad Hussain. A game theory based trust model for Vehicular Ad hoc Networks (VANETs). *Computer Networks*, 121:152–172, **2017**. doi:10.1016/j.comnet.2017.04.024.

[16] Honghao Gao, Can Liu, Yuyu Yin, Yueshen Xu, and Yu Li. A hybrid approach to trust node assessment and management for VANETs cooperative data communication: Historical interaction perspective. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):16504–16513, **2021**. doi:10.1109/TITS.2021.3129458.

[17] Matthias Grossglauser and David NC Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM transactions on networking*, 10(4):477–486, **2002**. doi:10.1109/TNET.2002.801403.

[18] CS Odessa. ConceptDraw sample diagrams created using the Local Vehicular Networking library from the Vehicular Networking solution. `https://www.conceptdraw.com/solution-park/vehicular-networking/`. Last accessed 31 Jan 2023.

[19] Cory Cornelius, Apu Kapadia, David Kotz, Dan Peebles, Minho Shin, and Nikos Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 211–224. **2008**. doi:10.1145/1378600.1378624.

[20] Michael I-C Wang, Charles H-P Wen, and H Jonathan Chao. Hierarchical cooperation and load balancing for scalable autonomous vehicle routing in multi-access edge computing environment. *IEEE Transactions on Vehicular Technology*, **2023**. doi:10.1109/TVT.2023.3236783.

[21] Haoyang Che, Yucong Duan, Chen Li, and Lei Yu. On trust management in vehicular ad hoc networks: A comprehensive review. *Frontiers in The Internet of Things*, 1:995233, **2022**. doi:10.3389/friot.2022.995233.

[22] Surbhi Sharma and Baijnath Kaushik. A survey on internet of vehicles: Applications, security issues & solutions. *Vehicular Communications*, 20:100182, **2019**. doi:10.1016/j.vehcom.2019.100182.

[23] Shukor Abd Razak, SM Furnell, Nathan L Clarke, and Phillip J Brooke. Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks. *Ad Hoc Networks*, 6(7):1151–1167, **2008**. doi:10.1016/j.adhoc.2007.11.004.

[24] Sunilkumar S Manvi and Shrikant Tangade. A survey on authentication schemes in VANETs for secured communication. *Vehicular Communications*, 9:19–30, **2017**. doi:10.1016/j.vehcom.2017.02.001.

[25] Sevil Sen and John Andrew Clark. Intrusion detection in mobile ad hoc networks. *Guide to Wireless Ad Hoc Networks*, pages 427–454, **2009**. doi:10.1007/978-1-84800-328-6_17.

[26] Han Yu, Zhiqi Shen, Chunyan Miao, Cyril Leung, and Dusit Niyato. A survey of trust and reputation management systems in wireless communications. *Proceedings of the IEEE*, 98(10):1755–1772, **2010**. doi:10.1109/JPROC.2010.2059690.

[27] Jin-Hee Cho, Ananthram Swami, and Ray Chen. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 13(4):562–583, **2011**. doi:10.1109/SURV.2011.092110.00088.

[28] Shuo Ma, Ouri Wolfson, and Jie Lin. A survey on trust management for intelligent transportation system. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Computational Transportation Science*, pages 18–23. ACM, **2011**. doi:10.1145/2068984.2068988.

[29] Jie Zhang. A survey on trust management for VANETs. In *2011 IEEE International Conference on Advanced Information Networking and Applications*, pages 105–112. IEEE, **2011**. doi:10.1109/AINA.2011.86.

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

[31] William Joseph Adams, George C Hadjichristofi, and NJ Davis. Calculating a node's reputation in a mobile ad hoc network. In *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, pages 303–307. IEEE, **2005**. doi:10.1109/PCCC.2005.1460573.

[32] Laurent Eschenauer, Virgil D Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *Security Protocols: 10th International Workshop, Cambridge, UK, April 17-19, 2002. Revised Papers 10*, pages 47–66. Springer, **2004**. doi:10.1007/978-3-540-39871-4_6.

[33] Bharat Bhargava, Leszek Lilien, Arnon Rosenthal, Marianne Winslett, Morris Sloman, Tharam S Dillon, Elizabeth Chang, Farookh Khadeer Hussain, Wolfgang Nejdl, Daniel Olmedilla, and Vipul Kashyap. The pudding of trust [intelligent systems]. *IEEE Intelligent Systems*, 19(5):74–88, **2004**. doi:10.1109/MIS.2004.52.

[34] Alfarez Abdul-Rahman and Stephen Hailes. Using recommendations for managing trust in distributed systems. In *Proceedings IEEE Malaysia International Conference on Communication*, volume 97. Citeseer, **1997**.

[35] Yan Lindsay Sun, Wei Yu, Zhu Han, and KJ Ray Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):305–317, **2006**. doi:10.1109/JSAC.2005.861389.

[36] Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 26(1):29–41, **1996**. doi:10.1109/3477.484436.

[37] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, **1995**. doi:10.1109/ICNN.1995.488968.

[38] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3):459–471, **2007**. doi:10.1007/s10898-007-9149-x.

[39] Shahrzad Saremi, Seyedali Mirjalili, and Andrew Lewis. Grasshopper optimisation algorithm: theory and application. *Advances in engineering software*, 105:30–47, **2017**. doi:10.1016/j.advengsoft.2017.01.004.

[40] Nichael Lynn Cramer. A representation for the adaptive generation of simple sequential programs. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 183–187. **1985**.

[41] John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. The MIT Press, Cambridge, MA, **1992**.

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

[43] John R Koza. Hierarchical genetic algorithms operating on populations of computer programs. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, volume 1, pages 768–774. **1989**.

[44] Riccardo Poli, William B Langdon, and Nicholas Freitag McPhee. *A Field Guide to Genetic Programming*. Published at `http://www.gp-field-guide.org.uk/`, **2008**. (With contributions by John R. Koza).

[45] Lee Spector. Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 401–408. **2012**. doi:10.1145/2330784.2330846.

[46] Brad L Miller and David E Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, **1995**.

[47] James Edward Baker. Adaptive selection methods for genetic algorithms. In *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pages 101–111. **1985**.

[48] James Edward Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, volume 206, pages 14–21. **1987**.

[49] John H Holland. *Adaptation in natural and artificial systems*. The University of Michigan Press, **1975**.

[50] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. The MIT Press, **1992**.

[51] Melanie Mitchell. *An introduction to genetic algorithms*. The MIT Press, **1998**. ISBN 9780262280013. doi:10.7551/mitpress/3927.001.0001.

[52] J Maynard Smith and George R Price. The logic of animal conflict. *Nature*, 246(5427):15–18, **1973**. doi:10.1038/246015a0.

[53] Conor Ryan, John James Collins, and Michael O Neill. Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic Programming: First European Workshop, EuroGP'98 Paris, France, April 14–15, 1998 Proceedings 1*, pages 83–96. Springer, **1998**. doi:10.1007/BFb0055930.

[54] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5:115–133, **1943**. doi:10.1007/BF02478259.

[55] Leandro Nunes De Castro and Jonathan Timmis. *Artificial immune systems: a new computational intelligence approach*. Springer Science & Business Media, **2002**.

[56] Trung Thanh Nguyen. *Continuous dynamic optimisation using evolutionary algorithms*. Ph.D. thesis, University of Birmingham, **2011**. https://etheses.bham.ac.uk/id/eprint/1296/.

[57] Yaochu Jin and Jürgen Branke. Evolutionary optimization in uncertain environments-a survey. *IEEE Transactions on evolutionary computation*, 9(3):303–317, **2005**. doi:10.1109/TEVC.2005.846356.

[58] Carlos Cruz, Juan R González, and David A Pelta. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15:1427–1448, **2011**. doi:10.1007/s00500-010-0681-0.

[59] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, **1997**. doi:10.1109/4235.585893.

[60] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A survey of evolutionary continuous dynamic optimization over two decades—part a. *IEEE Transactions on Evolutionary Computation*, 25(4):609–629, **2021**. doi:10.1109/TEVC.2021.3060014.

[61] Danial Yazdani, Ran Cheng, Cheng He, and Jürgen Branke. Adaptive control of subpopulations in evolutionary dynamic optimization. *IEEE Transactions on Cybernetics*, 52(7):6476–6489, **2020**. doi:10.1109/TCYB.2020.3036100.

[62] Aljawharah Alnasser, Hongjian Sun, and Jing Jiang. Cyber security challenges and solutions for V2X communications: A survey. *Computer Networks*, 151:52–67, **2019**. doi:10.1016/j.comnet.2018.12.018.

[63] Avleen Kaur Malhi, Shalini Batra, and Husanbir Singh Pannu. Security of vehicular ad-hoc networks: A comprehensive survey. *Computers & Security*, 89:101664, **2020**. doi:10.1016/j.cose.2019.101664.

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

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

[66] Cornelis Joost Van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, **1979**. ISBN 0408709294. `http://www.dcs.gla.ac.uk/Keith/Preface.html` Last accessed 31 Jan 2023.

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

[68] Xuanxia Yao, Xinlei Zhang, Huansheng Ning, and Pengjian Li. Using trust model to ensure reliable data acquisition in VANETs. *Ad Hoc Networks*, 55:107–118, **2017**. doi:10.1016/j.adhoc.2016.10.011.

[69] Mingshun Sun, Ming Li, and Ryan Gerdes. A data trust framework for VANETs enabling false data detection and secure vehicle tracking. In *2017 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, **2017**. doi:10.1109/CNS.2017.8228654.

[70] Seyed Ahmad Soleymani, Abdul Hanan Abdullah, Mahdi Zareei, Mohammad Hossein Anisi, Cesar Vargas-Rosales, Muhammad Khurram Khan, and Shidrokh Goudarzi. A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing. *IEEE Access*, 5:15619–15629, **2017**. doi:10.1109/ACCESS.2017.2733225.

[71] Hesham El Sayed, Sherali Zeadally, and Deepak Puthal. Design and evaluation of a novel hierarchical trust assessment approach for vehicular networks. *Vehicular Communications*, 24:100227, **2020**. doi:10.1016/j.vehcom.2019.100227.

[72] Jinsong Zhang, Kangfeng Zheng, Dongmei Zhang, and Bo Yan. AATMS: An anti-attack trust management scheme in VANET. *IEEE Access*, 8:21077–21090, **2020**. doi:10.1109/ACCESS.2020.2966747.

[73] Abir Mchergui, Tarek Moulahi, and Sherali Zeadally. Survey on artificial intelligence (AI) techniques for vehicular ad-hoc networks (VANETs). *Vehicular Communications*, 34:100403, **2022**. doi:10.1016/j.vehcom.2021.100403.

[74] Erfan A Shams, Ahmet Rizaner, and Ali Hakan Ulusoy. Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks. *Computers & Security*, 78:245–254, **2018**. doi:10.1016/j.cose.2018.06.008.

[75] Na Fan and Chase Q Wu. On trust models for communication security in vehicular ad-hoc networks. *Ad Hoc Networks*, 90:101740, **2019**. doi:10.1016/j.adhoc.2018.08.010.

[76] Dajun Zhang, F Richard Yu, Ruizhe Yang, and Li 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.

[77] Saqib Hakak, Thippa Reddy Gadekallu, Swarna Priya Ramu, Praveen Kumar Reddy Maddikunta, Chamitha de Alwis, Madhusanka Liyanage, et al. Autonomous vehicles in 5g and beyond: A survey. *arXiv preprint arXiv:2207.10510*, **2022**. doi:10.48550/arXiv.2207.10510.

[78] Chandrasekar Ravi, Anmol Tigga, G Thippa Reddy, Saqib Hakak, and Mamoun Alazab. Driver identification using optimized deep learning model in smart transportation. *ACM Transactions on Internet Technology*, 22(4):1–17, **2022**. doi:10.1145/3412353.

[79] Mehmet Aslan and Sevil Sen. Evolving trust formula to evaluate data trustworthiness in VANETs using genetic programming. In *Applications of Evolutionary Computation: 22nd International Conference, EvoApplications 2019, Held as Part of EvoStar 2019, Leipzig, Germany, April 24–26, 2019, Proceedings 22. Springer International Publishing*, pages 413–429. Springer, **2019**. doi:10.1007/978-3-030-16692-2_28.

[80] Pranav Kumar Singh, Shivam Gupta, Ritveeka Vashistha, Sunit Kumar Nandi, and Sukumar Nandi. Machine learning based approach to detect position falsification attack in vanets. In *International Conference on Security & Privacy*, pages 166–178. Springer, **2019**. doi:10.1007/978-981-13-7561-3_13.

[81] Bernabé Dorronsoro, Patricia Ruiz, Grégoire Danoy, Yoann Pigné, and Pascal Bouvry. *Survey on Optimization Problems for Mobile Ad Hoc Networks*, chapter 3, pages 49–78. John Wiley & Sons, **2014**. doi:10.1002/9781118833209.ch3.

[82] Daniel G Reina, Patricia Ruiz, R Ciobanu, SL Toral, Bernabé Dorronsoro, and Ciprian 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):2082496, **2016**. doi:10.1155/2016/2082496.

[83] Vasiliy Krundyshev, Maxim Kalinin, and Peter Zegzhda. Artificial swarm algorithm for VANET protection against routing attacks. In *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pages 795–800. IEEE, **2018**. doi:10.1109/ICPHYS.2018.8390808.

[84] Ankit Kumar, Pankaj Dadheech, Rajani Kumari, and Vijander Singh. An enhanced energy efficient routing protocol for VANET using special cross over in genetic algorithm. *Journal of Statistics and Management Systems*, 22(7):1349–1364, **2019**. doi:10.1080/09720510.2019.1618519.

[85] Alagan Ramasamy Rajeswari, Kanagasabai Kulothungan, Sannasi Ganapathy, and Arputharaj Kannan. A trusted fuzzy based stable and secure routing algorithm for effective communication in mobile adhoc networks. *Peer-to-Peer Networking and Applications*, 12(5):1076–1096, **2019**. doi:10.1007/s12083-019-00766-8.

[86] Youcef Azzoug and Abdelmadjid Boukra. Bio-inspired VANET routing optimization: an overview. *Artificial Intelligence Review*, 54(2):1005–1062, **2021**. doi:10.1007/s10462-020-09868-9.

[87] DG Reina, Radu-Ioan Ciobanu, SL Toral, and C Dobre. A multi-objective optimization of data dissemination in delay tolerant networks. *Expert Systems with Applications*, 57:178–191, **2016**. doi:10.1016/j.eswa.2016.03.038.

[88] Muhammet Ali Karabulut, AFM Shahen Shah, and Haci Ilhan. Performance optimization by using artificial neural network algorithms in VANETs. In *2019 42nd International conference on telecommunications and signal processing (TSP)*, pages 633–636. IEEE, **2019**. doi:10.1109/TSP.2019.8768830.

[89] Vishakha Chourasia, Sudhakar Pandey, and Sanjay Kumar. Optimizing the performance of vehicular delay tolerant networks using multi-objective PSO and artificial intelligence. *Computer Communications*, 177:10–23, **2021**. doi:10.1016/j.comcom.2021.06.006.

[90] Janusz Kusyk, M Umit Uyar, and Cem Safak Sahin. Survey on evolutionary computation methods for cybersecurity of mobile ad hoc networks. *Evolutionary Intelligence*, 10(3):95–117, **2018**. doi:10.1007/s12065-018-0154-4.

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

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

[93] Shubhra Dwivedi, Manu Vardhan, Sarsij Tripathi, and Alok Kumar Shukla. Implementation of adaptive scheme in evolutionary technique for anomaly-based intrusion detection. *Evolutionary Intelligence*, 13(1):103–117, **2020**. doi:10.1007/s12065-019-00293-8.

[94] Ankit Thakkar and Ritika Lohiya. Role of swarm and evolutionary algorithms for intrusion detection system: A survey. *Swarm and evolutionary computation*, 53:100631, **2020**. doi:10.1016/j.swevo.2019.100631.

[95] RS Raghav, U Prabu, M Rajeswari, D Saravanan, and Kalaipriyan Thirugnanasambandam. Cuddle death algorithm using ABC for detecting unhealthy nodes in wireless sensor networks. *Evolutionary Intelligence*, pages 1–13, **2021**. doi:10.1007/s12065-021-00570-5.

[96] Ugur Eray Tahta, Sevil Sen, and Ahmet Burak Can. GenTrust: A genetic trust management model for peer-to-peer systems. *Applied Soft Computing*, 34:693–704, **2015**. doi:10.1016/j.asoc.2015.04.053.

[97] CK Shyamala, Niveda Ashok, and Bhavya Narayanan. A decimal coded genetic algorithm recommender for P2P systems. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pages 335–348. Springer, **2017**. doi:10.1007/978-981-10-3174-8_30.

[98] W Yuan and D Guan. Optimized trust-aware recommender system using genetic algorithm. *Neural Network World*, 27(1):77, **2017**. doi:10.14311/NNW.2017.27.004.

[99] Ajay Vikram Singh, Vandana Juyal, and Ravish Saggar. Trust based intelligent routing algorithm for delay tolerant network using artificial neural network. *Wireless Networks*, 23(3):693–702, **2017**. doi:10.1007/s11276-015-1166-y.

[100] Grantej Vinod Otari and Vijay Ram Ghorpade. A trust management model based on NSGA-II in mobile grid system. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 24(3):235–242, **2020**. doi:10.3233/KES-200045.

[101] Kutub Thakur and Gulshan Kumar. Nature Inspired Techniques and Applications in Intrusion Detection Systems: Recent Progress and Updated Perspective. *Archives of Computational Methods in Engineering*, 28(4):2897–2919, **2021**. doi:10.1007/s11831-020-09481-7.

[102] Surbhi Sharma and Baijnath Kaushik. A survey on nature-inspired algorithms and its applications in the Internet of Vehicles. *International Journal of Communication Systems*, 34(12):e4895, **2021**. doi:10.1002/dac.4895.

[103] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A survey of evolutionary continuous dynamic optimization over two decades—part b. *IEEE Transactions on Evolutionary Computation*, 25(4):630–650, **2021**. doi:10.1109/TEVC.2021.3060012.

[104] Shengxiang Yang, Hui Cheng, and Fang Wang. Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):52–63, **2009**. doi:10.1109/TSMCC.2009.2023676.

[105] Hui Cheng and Shengxiang 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, pages 562–571. Springer, **2010**. doi:10.1007/978-3-642-12239-2_58.

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

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

[108] Hui Cheng and Shengxiang Yang. Genetic algorithms for dynamic routing problems in mobile ad hoc networks. In *Evolutionary Computation for Dynamic Optimization Problems*, pages 343–375. Springer, **2013**. doi:10.1007/978-3-642-38416-5_14.

[109] Darren M Chitty and Marcel L Hernandez. A hybrid ant colony optimisation technique for dynamic vehicle routing. In *Genetic and Evolutionary Computation Conference*, pages 48–59. Springer, **2004**. doi:10.1007/978-3-540-24854-5_5.

[110] Li-Ning Xing, Philipp Rohlfshagen, Ying-Wu Chen, and Xin 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):1110–1123, **2011**. doi:10.1109/TSMCB.2011.2107899.

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

[112] Muhammad Sameer Sheikh, Jun Liang, and Wensong Wang. Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey. *Wireless Communications and Mobile Computing*, 2020:1–25, **2020**. doi:10.1155/2020/5129620.

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

[114] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12(6):e0177678, **2017**. doi:10.1371/journal.pone.0177678.

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

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

[117] Davide Chicco, Niklas Tötsch, and Giuseppe 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):1–22, **2021**. doi:10.1186/s13040-021-00244-z.

[118] Xiaohui Hu and Russell 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)*, volume 2, pages 1666–1670. IEEE, **2002**. doi:10.1109/CEC.2002.1004492.

[119] Xiaodong Li, Jürgen Branke, and Tim Blackwell. Particle swarm with speciation and adaptation in a dynamic environment. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 51–58. **2006**. doi:10.1145/1143997.1144005.

[120] Gregory R Kramer and John C Gallagher. Improvements to the *CGA enabling online intrinsic evolution in compact EH devices. In *NASA/DoD Conference on Evolvable Hardware, 2003. Proceedings.*, pages 225–231. IEEE, **2003**. doi:10.1109/EH.2003.1217670.

[121] Marius Riekert, Katherine M Malan, and AP Engelbrect. Adaptive genetic programming for dynamic classification problems. In *2009 IEEE congress on evolutionary computation*, pages 674–681. IEEE, **2009**. doi:10.1109/CEC.2009.4983010.

[122] The NS-3 Consortium. NS-3 a discrete-event network simulator for internet systems. `https://www.nsnam.org/`, **2008**. Last accessed 31 Jan 2023.

[123] Sean Luke. ECJ A Java-based Evolutionary Computation Research System. `https://cs.gmu.edu/~eclab/projects/ecj/`, **1998**. Last accessed 31 Jan 2023.

[124] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile computing*, pages 153–181, **1996**. doi:10.1007/978-0-585-29603-6_5.

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

[126] Valery Naumov, Rainer Baumann, and Thomas 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*, pages 108–119. ACM, **2006**. doi:10.1145/1132905.1132918.

[127] Elaheh Biglar Beigi, Hossein Hadian Jazi, Natalia Stakhanova, and Ali A Ghorbani. Towards effective feature selection in machine learning-based botnet detection approaches. In *2014 IEEE Conference on Communications and Network Security*, pages 247–255. IEEE, **2014**. doi:10.1109/CNS.2014.6997492.

[128] Shi-Yong Chen, Yang Yu, Qing Da, Jun Tan, Hai-Kuan Huang, and Hai-Hong 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*, pages 1187–1196. **2018**. doi:10.1145/3219819.3220122.

[129] Jingwen Wang, Xuyang Jing, Zheng Yan, Yulong Fu, Witold Pedrycz, and Laurence T Yang. A survey on trust evaluation based on machine learning. *ACM Computing Surveys (CSUR)*, 53(5):1–36, **2020**. doi:10.1145/3408292.