Security prediction and forecasting for trust management systems in VANETs

by

Mohammed Abdelmaguid

A thesis submitted to the
School of Computing
in conformity with the requirements for
the Degree of Doctor of Philosophy

Queen's University
Kingston, Ontario, Canada
September 2024

Abstract

This thesis addresses the enhancement of proactive security measures in Vehicular Ad-hoc Networks (VANETs), which are crucial for enhancing road safety and neutralizing cyber threats that can compromise human safety and network efficacy. Despite VANETs' potential to improve road safety, the predominance of reactive security measures is insufficient to combat the evolving landscape of cyber threats, and current datasets in VANETs' research are inadequate for evaluating proactive security approaches, thus stifling advancement in this crucial area. This research highlights the necessity of shifting from traditional reactive security strategies to proactive measures that anticipate and mitigate unforeseen attacks, a significant challenge with current methodologies that generally focus on known threats. Trust Management Systems (TMS) have proven effective against post-authenticated nodes. Trust is the level of confidence in accepting and acting on the received information. Trust in TMS is comprised of three components: Trust Subject, Trust Service, and Trust Origin. To enhance the detection of misbehavior messages in VANETs, we incorporate situational awareness (SA) to forecast the trustworthiness of Trust Subjects. To predict the potential impact of attacks on Trust Services, we make several contributions. We divide the attack life cycle of VANETs into proactive and retroactive phases. The

proactive phase addresses the attack endgame, which refers to the impact of the attack on the physical world, such as causing accidents or creating hazards. We design an approach to predict and mitigate the attack endgame in VANETs. Furthermore, we leverage attack rate data from Trust Origin and introduce a new approach using honeypots to evaluate the network's security status by examining its ability to identify known attacks, referred to as prepared-for attacks. This information lays the groundwork to predict future attacks, which we refer to as unprepared-for attacks. Given that current datasets in VANET research are inadequate for evaluating proactive security approaches, thus impeding advancement in this crucial area, we present a new dataset, VeReMi for Attack Prediction (VeReMiAP), to evaluate the impact of attacks. This dataset aims to address the gaps in existing proactive security research tools and foster further development in proactive security strategies within VANETs. Implementation and evaluation of these techniques, utilizing open shared datasets and frameworks like VeReMi and F2MD, demonstrate the effectiveness in proactively predicting and mitigating the effects of attacks.

Co-Authorship

1. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "Attack Endgame: Proactive Security Approach for Predicting Attack Consequences in VANET." *ICC 2023-IEEE International Conference on Communications.*, 2023, pp. 3762–3767.

2. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "A VeReMi-based Dataset for Predicting the Effect of Attacks in VANETs." *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems.* 2023, p. 175–179.

3. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "Samm: Situation awareness with machine learning for misbehavior detection in vanet." *Proceedings of the 17th International Conference on Availability, Reliability and Security.* 2022, pp. 1–10.

4. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "Beyond the Attack: Strategies for Attack Endgame Mitigation in VANETs." *IEEE Transactions on Vehicular Technology.* 2024, Submitted.

5. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "Securing the Unforeseen: Enhancing VANET Security with Dynamic

Honeypots and Attack Rate Analysis." *Vehicular Communications.* 2024, Submitted.

6. Mohammed A. Abdelmaguid, Hossam S. Hassanein, and Mohammad Zulkernine. "Poster: Security Prediction and Forecasting for a Trust Management System in VANET." *CCECE 2024-Canadian Conference on Electrical and Computer Engineering.* 2024.

Acknowledgments

"Whoever does not thank people has not thanked Allah" (Prophet Muhammad PBUH)

First and foremost, I would like to express my deepest gratitude to Allah (SWT) for granting me the strength, patience, and perseverance to complete this thesis.

I am profoundly grateful to my supervisors, Prof. Hossam Hassanein and Prof. Mohammad Zulkernine, for their endless support, guidance, and invaluable teachings. Their mentorship has not only been instrumental in the successful completion of this thesis but has also significantly contributed to my professional career and personal development.

I extend my heartfelt thanks to my parents, brother, and sister, whose unwavering support and love have been my anchor. Despite the physical distance between us, their presence has always been felt, and their encouragement has motivated me to persevere.

Finally, I would like to thank my colleagues and friends for their helpful discussions and emotional support. Their camaraderie and insights have enriched my research experience. I would also like to extend my gratitude to Basia Palmer for her assistance in reviewing my thesis.

Thank you all for your profound contributions and support.

Statement of Originality

I hereby certify that this Ph.D. thesis is an original work and that all ideas and inventions credited to others have been appropriately referenced.

Mohammed Abdelmaguid

September, 2024

Contents

List of Tables

List of Figures

List of Abbreviations

ANN Artificial neural network.

ARIMA Auto-regressive integrated moving average.

AV Autonomous vehicle.

BP Backpropagation.

BRNN-LSTM Bi-directional recurrent neural networks with long short-term memory.

BSMs Basic Safety Messages.

C-V2X Cellular-Vehicle-to-Everything.

CA Certificate authorities.

CAMs Cooperative awareness messages.

CAN Control area network.

DBM Dynamic blacklist management.

DBSCAN Density-based spatial clustering of applications with noise.

DoS Denial of service.

DPoS Delegated proof-of-stake.

DSRC Dedicated short-range communications.

ECUs Electronic control units.

EM Emergency message.

EPE Endgame prevention edge.

EPL Endgame prevention lite.

F2MD Framework for misbehavior detection.

FL Federated learning.

FRA Fake reporting attack.

GARCH Generalized autoregressive conditional heteroscedasticity.

GDELT Global Database of Events, Language, and Tone.

GEM GDELT event mentions.

GET GDELT event tone.

GIS Geographic information system.

GNSS Global navigation satellite system.

HNA Hackmageddon number of attacks.

HPIDS Honeypot optimized intrusion detection system.

HPSD Honeypot selfishness detection.

IDPS Intrusion detection and prevention systems.

IDS Intrusion detection systems.

IMU Inertial measurement unit.

IPS Intrusion prediction system.

IRP Incident response planning.

ITS Intelligent transportation system.

LT Logistic trust.

MANETs Mobile ad-hoc networks.

MDS Misbehavior detection system.

ML Machine learning.

NN Neural network.

OBU Onboard unit.

OSNs Online social networks.

PAP Prepared-for attack predictor.

PKI Public key infrastructure.

PSIDs Provider service identifiers.

QoS Quality of service.

RBF Radial basis Function.

ReLU Rectified linear unit.

RNN Recurrent neural network.

RREQ Route request.

RSMA Reputation system-based lightweight message authentication.

RSSI Received signal strength indicator.

RSU Roadside unit.

SA Situational awareness.

SAMM Situation awareness with machine learning for misbehavior detection in VANET.

SART Subjective ranking technique.

SAT Situational awareness trust architecture.

SDN Software-defined networking.

SL Supervised learning.

SMC Scalable modulation and coding.

SQL Structured Qquery language.

SQLIA Structured query language injection attack.

SUMO Simulation of urban mobility.

SVM Support vector machine.

TA Trusted authority.

TAM Twitter attack mentions.

TCSVM Two-class support vector machine.

TLC Time/location-critical.

TMS Trust management systems.

TraCI Traffic control interface.

TTP Trusted third party.

UPAP Unprepared-for attack predictor.

VANETs Vehicular ad-hoc networks.

WAVE Wireless access in vehicular environments.

WSMP WAVE Short Message Protocol.

Chapter 1

Introduction

The rise of Intelligent Transportation System (ITS) represents a significant advancement in integrating information technologies into the transportation sector [1]. Recognized for its potential to revolutionize road safety and collision prevention, ITS has garnered substantial investment from numerous governments. According to predictions, the global ITS investment is expected to reach USD 68 billion by 2026 [2]. This investment highlights the increasing importance of these systems in modern transportation infrastructure.

The success of ITS depends mainly on the development of the Vehicular ad-hoc Networks (VANETs), which is considered a critical component for ensuring the reliability of ITS [3]. VANETs are self-organizing networks composed of vehicles, each outfitted with Electronic Control Units (ECUs). These vehicles communicate in a decentralized manner, significantly enhancing the driving experience. This improvement is evidenced through increased safety and more efficient traffic management [4]. These networks employ various forms of communication, including Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) [5], with each type contributing uniquely to the network's effectiveness.

However, the proliferation of connected vehicles within VANETs introduces critical security concerns [5, 6]. Since messages in VANETs are broadcast to all network participants, it is crucial to have robust mechanisms to distinguish between genuine and malicious communications. Ensuring message authenticity and accuracy is paramount to maintaining secure and safe network communication. While standard protocols, such as those outlined in IEEE 1609 Family of Standards for Wireless Access in Vehicular Environments [7], address message authentication, ensuring message correctness remains challenging. The presence of malicious entities within the network or the transmission of false information by drivers, referred to as misbehavior, can compromise the integrity of the entire communication process.

Traditional security approaches predominantly focus on addressing the consequences of attacks, limiting the scope of potential responses and preventive measures. Consequently, the ability to predict and forecast attacks has become increasingly vital [8]. *Attack forecasting* relies on situational awareness and utilizes historical data to make predictions. This process involves synthesizing recent trends with past events to generate specific, definitive outcomes. In contrast, *attack prediction* is an estimation process based on factual evidence and intuitive judgment, aiming to make probabilistic statements about future events. This distinction highlights the nuanced complexities in developing effective security strategies for ITS and VANETs.

1.1 Motivation and Objectives

In the domain of VANETs, security is divided into two main categories, as illustrated in Figure 1.1. The first category is infrastructure-related security, involving Trusted Authorities (TA), Trusted Third Parties (TTP), Certificate Authorities (CA), and

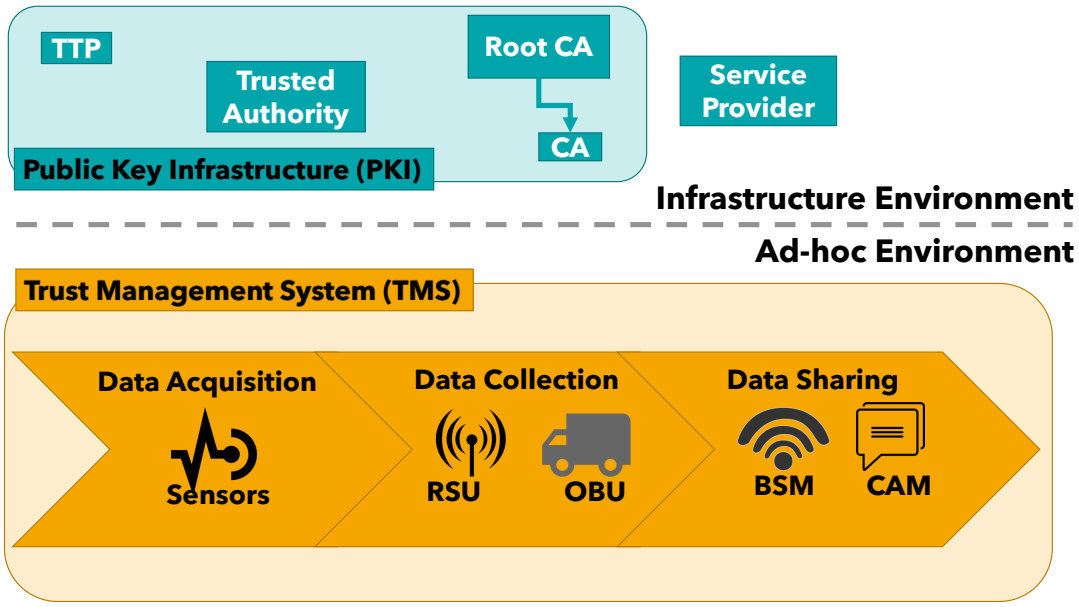


Figure 1.1: Comparative overview of security in VANET infrastructure environment and ad-hoc environment

service providers, especially relevant in cellular V2X communication. These entities are responsible for key management, certificate issuance, and ensuring secure communication channels. Infrastructure security is strongly supported by Public Key Infrastructure (PKI), which efficiently handles the authentication, issuance, and revocation of certificates for network participants. The second category pertains to ad-hoc security, focusing on the network entities equipped with sensors that collect data. This data is shared between On-Board Units (OBUs) and Road-Side Units (RSUs) through Basic Safety Messages (BSMs) or Cooperative Awareness Messages (CAMs). The security of the ad-hoc segment of VANETs deals with insider attacks, where authenticated entities broadcast misleading information. These threats originate from malicious entities that bypass PKI authentication or from legitimate users who act against the network's interests. This type of attack highlights the critical

role of Trust Management Systems (TMS) in ensuring the integrity of the information exchanged within VANETs.

The challenge of insider attacks in VANETs is notably complex due to the difficulty in detecting and mitigating threats from entities considered part of the system. The unpredictability of such attacks exacerbates the situation, demonstrating the insufficiency of traditional defenses and underscoring the critical need for more sophisticated proactive security measures. This is particularly important not only for the integrity of the network but also for preventing dangerous situations on the road, such as accidents or hazards, making the confrontation of insider threats a paramount endeavor.

The work presented in this thesis is motivated by the significant gap in existing studies on proactive security measures and trust-based solutions suitable for VANETs' unique requirements. Therefore, the primary goal of this research is to develop a novel proactive trust management system designed for the early detection and mitigation of insider misbehavior in VANETs. By focusing on this goal, this research seeks to improve the security of VANETs, ensuring the integrity of network communications and, ultimately, the safety of road users.

In developing the proactive trust management system, our objective is to address the core components of trust management systems. As illustrated in Figure 1.2, the system is structured around three pivotal modules, each corresponding to a component of trust management, which is integrated with a proactive security approach. The modules within the proposed approach include (1) the implementation of situational awareness for evaluating trust subjects, (2) the application of attack endgame

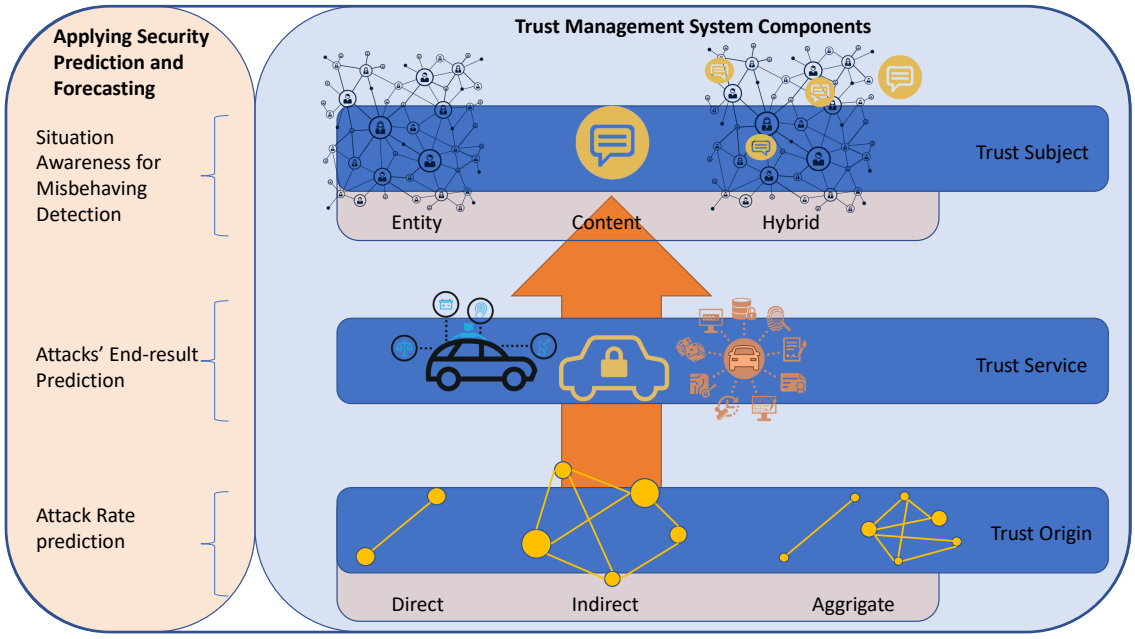


Figure 1.2: A trust management system based on attack prediction and forecasting for VANETs

prediction methodologies for trust services, and (3) the incorporation of attack prediction strategies concerning trust origin. These modules encompass our research objectives to construct a proactive trust management system compatible with VANETs environments, enhancing their security posture. This approach signifies the system's ambition to establish a secure and proactive trust management system tailored for VANETs.

1.2 Research Statement

In the domain of VANETs, the distinction between security prediction and detection remains a subject of considerable debate within the research community. While literature exists on security prediction and forecasting in VANETs [9, 10], a notable

ambiguity persists in differentiating prediction from detection [11, 12]. Many existing studies have inadvertently conflated these two concepts, categorizing detection methodologies as predictive solutions [13]. Conversely, there are instances where predictive solutions are proposed without employing a clearly defined metric to assess the efficacy of proactive strategies.

We believe that:
This research emphasizes the critical importance of establishing attack prediction in VANETs, clearly distinguishing it from attack detection. The goal is to enhance its utility and develop more secure, timely solutions, thereby streamlining and solidifying the foundation for future research in this field.

1.3 Contributions

The main contributions of this thesis are as follows:

- Employing situational awareness for trust subject and forecasting: We present an innovative framework incorporating situational awareness into misbehavior detection in VANETs [14]. Situational awareness refers to the ability to perceive, comprehend, and project the meaning of environmental elements and events in relation to time and space [15]. The proposed framework has two objectives: first, measuring situational awareness to depict the current state of the network, and second, leveraging this comprehensive understanding, combined with machine learning model insights, to enhance trust subject evaluation within the TMS. Furthermore, situational awareness is utilized as a key approach for forecasting, providing a predictive edge to the security mechanisms in VANETs. The research also contributes to standardizing the field by

setting new baseline metrics for one of the famous datasets in VANETs, the VeReMi dataset [16], thus offering future researchers a valuable reference point for advancing misbehavior detection methodologies.

- Utilizing attack endgame prediction for trust services: Recognizing the critical importance of security-related services in VANETs, we propose a novel method for predicting the impacts of misleading information shared within the network [17]. This predictive approach aids security authorities in VANETs in formulating proactive strategies to mitigate the effects of attacks. The research redefines the lifecycle of attacks on VANETs by including the consequences of these attacks, with results demonstrating the feasibility of predicting the impact of an attack minutes before its occurrence.

- Dataset for network attack prediction: Addressing the lack of datasets geared towards proactive security solutions in VANETs, we introduce VeReMiAP [18], a new dataset focusing on attack prediction rather than attack detection. The dataset extends the Framework For Misbehavior Detection (F2MD) by incorporating Cooperative Awareness Messages (CAMs), thereby enabling the investigation of new attack types. Additionally, a new class of attacks related to CAMs, the Fake Reporting Attack (FRA), is introduced. The dataset includes simulations of specific FRA scenarios, such as a vehicle reporting a false accident, and incorporates the effects of these attacks, such as creating a hazardous situation. Comprehensive temporal, geospatial, and behavioral analyses of the dataset provide a deeper understanding of the data.

- Predicting attack rate for trust origin: This component of the research focuses

on predicting the attack rate, which is instrumental in assessing the trustworthiness of both overall and sub-networks within VANETs. This prediction aids in the comprehensive evaluation of network security, enabling a more effective trust management approach.

1.4 Outline

The structure of this thesis is organized in the following manner.

Chapter 2 delves into the background and pertinent literature. It presents the significance, foundational technologies, and key principles of VANET security, encompassing aspects of attack prediction, forecasting, and trust management systems.

Chapter 3 introduces the concept of situational awareness and our methodology for its measurement, particularly in the context of insider misbehaving attacks in VANETs. This chapter provides an in-depth examination of the misbehavior detection model, integrating situational awareness, and concludes with an analysis of the simulation results obtained from the proposed model.

Chapter 4 discusses the development of a specialized dataset designed for proactive security measures in VANETs. This chapter details the extension of the F2MD framework, which facilitates the generation of this new dataset. It meticulously describes the setup of our simulation environment and offers a comprehensive account of the dataset generation process. The chapter concludes with a presentation of the simulation outcomes and a thorough analysis and evaluation of the entire process. This evaluation not only underscores the efficacy of the dataset but also its applicability in enhancing VANET security.

Chapter 5 highlights the attack life cycle within VANETs and its relation to the

concept of attack endgame. A discussion on countermeasures for attack endgame and strategies to mitigate attack effects is presented, followed by a detailed exposition of the systems, simulation settings, and the results.

Chapter 6 delves into the deployment of honeypots to quantify the attack rate within VANETs. It further utilizes the derived attack rates from anticipated attacks as a predictive tool for identifying the likelihood of unforeseen attacks within the network. A series of experiments are conducted to assess the efficacy of the proposed methodology. These experiments provide evidence supporting the deployment of honeypots in VANETs, offering strategic insights and recommendations for optimizing data collection on network attacks and enhancing the accuracy of attack rate calculation.

Chapter 7 summarizes thesis topics and discusses potential avenues for future research.

Chapter 2

Background

This chapter provides a comprehensive overview of the existing literature and foundational concepts pertinent to our research, laying the groundwork for understanding the current state of knowledge in the field of VANET security, with a particular emphasis on proactive security measures.

2.1 Vehicular Ad-hoc Networks (VANETs)

As shown in Figure 2.1, the architecture of VANETs encompasses interactions between the OBU installed in vehicles and the RSU, strategically placed along roadways. This synergy between the OBUs and RSUs yields enhanced outcomes in terms of data delivery, throughput, and overall network efficiency. Moreover, it is possible to include a TTP or TA in the VANET architecture. As a self-organized network, VANETs facilitate decentralized connectivity among vehicles, significantly improving the driving experience by improving safety and optimizing traffic management [4].

VANETs are integral to the functionality of ITS, offering benefits such as real-time communication, enhanced mobility, scalability, interoperability, and cost-effectiveness.

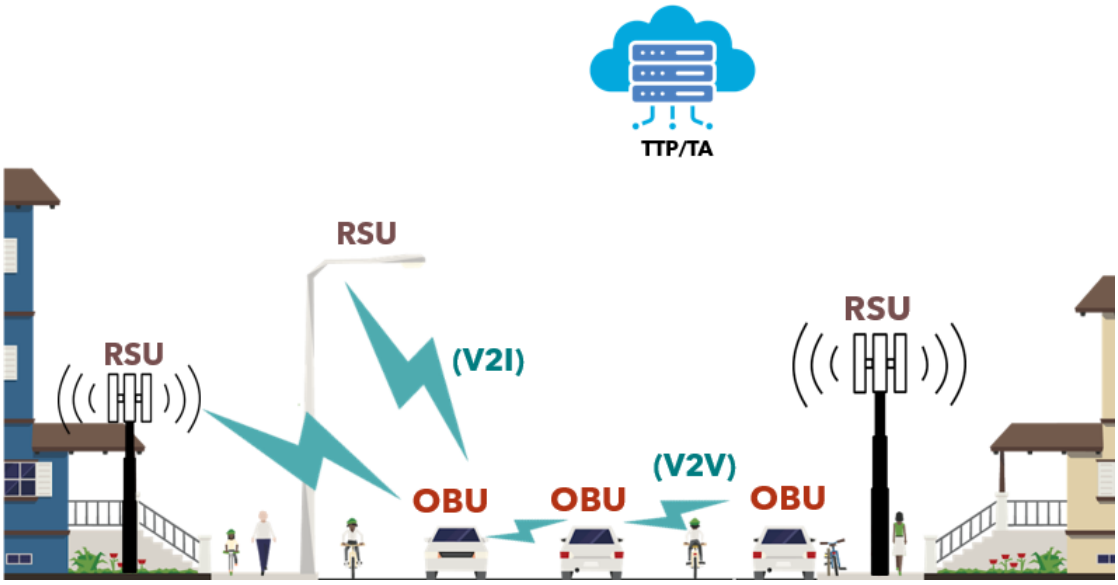


Figure 2.1: VANET system architecture

2.1.1 Exchanged Messages in VANETs

The IEEE Standards Association recognizes the use of a specialized radio communication system known as Wireless Access in Vehicular Environments (WAVE) for message exchange in VANETs. WAVE is designed to provide seamless, interoperable services tailored to transportation needs. The services within WAVE encompass a broad range of applications, protocols, and higher layer entities that operate above the WAVE Short Message Protocol (WSMP) layer.

WSMP, a highly efficient messaging protocol used in WAVE systems, facilitates communication through three types of messages: WAVE Short Messages (WSM), WAVE Service Advertisements (WSA), and Basic Safety Messages (BSM) [19], with CAMs [20] as a special version. WSMs are the primary messages utilized by WSMP, while WSAs are periodic messages employed by WAVE systems to advertise available services. Furthermore, each WSA may contain a list of Provider Service Identifiers

(PSIDs), which reside in the header of WSM messages transported by WSMP. These PSIDs are crucial as they indicate the services available on the network and provide the necessary information for receiving and processing the WSMs related to each advertised service. This intricate system of communication protocols and messaging formats underlines the sophistication of WAVE in enhancing the functionality and efficiency of VANETs.

Building on the foundational concepts of WAVE systems in VANETs, our research specifically concentrates on Basic Safety Messages (BSMs) and CAMs.

BSMs, as delineated in the Society of Automotive Engineers (SAE) J2735 standard [21], are pivotal in the exchange of information within VANETs. These messages, periodically transmitted between nearby vehicles, are instrumental in conveying critical data such as speed, location, and time [22]. The primary utility of BSMs lies in their application for collision avoidance and traffic jam detection. However, the inherent public nature of BSMs poses a significant vulnerability; malicious nodes within the network can exploit this openness to broadcast false information, leading to potentially catastrophic outcomes [23].

In conjunction with BSMs, CAMs play a vital role in the communication ecosystem of VANETs. These messages are triggered periodically and are crucial for short-distance communication between vehicles and RSUs. Typically, CAMs are transmitted in response to specific conditions such as change in speed or position which could reflects heavy traffic, road construction, or accidents. They provide not only real-time information about the vehicle but also encompass additional data pertinent to road conditions, traffic congestion, and incidents. The integration of CAMs and

BSMs within VANETs underscores the complexity and multifaceted nature of vehicular communication systems, necessitating robust security and trust mechanisms to ensure the reliability and safety of transmitted information.

2.1.2 Communication Technologies in VANETs

In VANETs, two primary technologies are at the forefront of facilitating communication: Dedicated Short-Range Communications (DSRC) and cellular networks, including LTE and 5G [24]. DSRC is particularly notable for its unique design, which is tailored to vehicular applications and operates within the 5.9 GHz spectrum with 75 MHz spectrum, and this enables it to provide low-latency, high-reliability communication essential for V2V and V2I exchanges [5]. The specialization of DSRC makes it highly suitable for safety-critical applications that require immediate response, such as collision avoidance and traffic signal management.

The communication technology used in the experiments is the DSRC technology. Despite evolving cellular technologies offering extensive coverage and catering to non-emergency functionalities such as entertainment systems, DSRC has retained its prominence in the VANET security landscape [25]. This is attributed to its steadfast reliability, the ability for direct communication bypassing the need for cellular infrastructure, and the dedicated spectrum allocation crucial for the exigencies of real-time, safety-oriented vehicular communication.

2.2 VANET Security

Security in connected vehicles has become a critical issue [5, 6]. In VANETs, messages are disseminated across the network, presenting a potential vulnerability where

counterfeit information can be shared indiscriminately. This necessitates a robust mechanism within VANET nodes to effectively discern between malicious and benign entities, as well as to validate the authenticity of the messages being exchanged [26].

Given the distinctive nature of VANETs, it becomes imperative to thoroughly investigate their unique characteristics and the attendant security challenges they present. This includes an analysis of how traditional attacks might be adapted to this new environment, as well as the anticipation of entirely new forms of attacks that could emerge. Understanding these aspects is crucial for developing advanced security protocols and safeguarding the integrity of communication within VANETs, thereby ensuring the safety and reliability of connected vehicle systems.

2.2.1 Security Characteristics in VANETs

VANETs exhibit distinct characteristics significantly influencing security dynamics. Firstly, the high mobility of nodes within VANETs leads to a rapidly changing network topology, rendering it an extremely active environment. This dynamism poses unique challenges for maintaining consistent and secure communication channels. Additionally, despite the unpredictable nature of individual vehicle movements, VANETs often display predictable motion patterns, such as those governed by traffic flows and road structures. This predictability can be a double-edged sword for security, offering potential predictability in traffic behavior and potentially making specific attacks more feasible.

The unbounded nature of VANETs' network size adds another layer of complexity, as it entails managing a vast and continuously evolving network without a fixed perimeter. This aspect challenges traditional security models, which typically rely

on well-defined network boundaries. Moreover, the essence of VANETs lies in their capacity for real-time, time-sensitive data exchange, which is critical for applications like collision avoidance and traffic management. However, the need for immediacy in communication also means that any delay caused by security protocols could have significant repercussions, necessitating a careful balance between security measures and the need for swift data transmission. These characteristics collectively shape the security landscape of VANETs, requiring tailored solutions that address their unique combination of high mobility, dynamic topology, predictable patterns, large scale, and time-critical communication needs.

2.2.2 Security Challenges in VANETs

VANETs present a unique set of security challenges, primarily influenced by their inherent operational requirements. One of the most critical aspects is the real-time constraint, which necessitates a transmission delay of no more than 100 milliseconds. This stringent requirement for rapid data exchange imposes significant constraints on the design and implementation of security protocols, as any process that introduces latency could undermine the network's effectiveness, particularly in time-sensitive scenarios such as collision avoidance.

Furthermore, VANETs face the challenge of data consistency liability. Even authenticated and deemed secure nodes can potentially engage in malicious activities. This risk underscores the necessity for continuous monitoring and dynamic security mechanisms that go beyond initial authentication and consider the ongoing behavior of network participants.

Lastly, the low tolerance for error in VANETs dictates that security protocols cannot rely on probabilistic designs commonly employed in other network types. While efficient in some contexts, probabilistic protocols may not guarantee the accuracy and reliability required in VANETs. This necessitates the development of security solutions that ensure precision and consistency, considering the high stakes involved in vehicular communication, where errors can have immediate and severe consequences. Collectively, these challenges highlight the need for robust, real-time, and reliable security strategies specifically tailored to the unique environment of VANETs.

2.2.3 Attacks in VANETs

In the context of VANETs, the nature and scope of attacks can be broadly categorized based on their targets, types, and the profiles of the attackers. Attacks within VANETs are generally aimed at two primary targets: inter-vehicle, which pertains to the network-related aspects, and intra-vehicle, which focuses on the vehicle-related components. This bifurcation underscores the dual nature of vulnerabilities in VANETs, encompassing both the communication network and the individual vehicular systems.

The types of attacks are classified into passive and active categories. Passive attacks involve the unauthorized interception or monitoring of network communications without altering the data, thereby posing a threat to the confidentiality of the information exchanged. Active attacks, on the other hand, involve the modification, fabrication, or disruption of data, directly impacting the integrity and availability of the network services.

The categorization of attackers in VANETs is also multifaceted, primarily distinguished as insiders versus outsiders and malicious versus rational attackers. Insider attackers are those who have legitimate access to VANETs and can exploit their privileged position, while outsider attackers are external entities without authorized access. Additionally, attackers can be characterized by their intent: malicious attackers aim to cause harm or disruption regardless of personal gain, whereas rational attackers are driven by specific objectives or benefits.

Understanding the diverse nature of attacks in VANETs is crucial for developing comprehensive security strategies. This involves not only addressing the technical vulnerabilities but also considering the varied motivations and access levels of potential attackers, thereby ensuring a robust defense mechanism tailored to the multifaceted threat landscape of VANETs.

2.2.4 Misbehaving Attacks in VANETs

The open sharing environment in VANETs led to dealing with a specific type of attack called misbehavior attack, where vehicles share misleading information. VANETs are particularly vulnerable to misbehaving and insider attacks compared to other networks. Misbehavior messages are authentic messages that contain false information [27], and exchanging such erroneous information in VANETs can propagate swiftly. Additionally, vehicle and driver profiling may depend heavily on this information. Thus, assessing the trustworthiness of the surrounding nodes became integral because we need to guarantee the correctness of the information received and protect the driver's decision-making process from being influenced by false information.

Misbehaving attackers can harm network operations by transmitting false or malicious information. In addition, insider attackers can access and spread sensitive information among network participants. Ensuring the security of VANETs, BSMs, and CAMs is essential for maintaining the reliability and trustworthiness of the information transmitted. Implementing security prediction methods can enhance the overall security and reliability of the network and its applications.

2.3 Reactive and Proactive Security

The traditional approach to security, often referred to as "reactive security," draws its roots from a 'castle model' strategy, focusing primarily on the detection and response to threats post-breach. This conventional paradigm in security management is typically considered an isolated operation, distinct from the core functionalities of a business or network system. However, its primary limitation lies in its narrow perspective of the threat landscape. Such an approach is increasingly seen as inadequate in the face of multifaceted and omnipresent threats, which demand more extensive measures for network protection.

Despite its limitations, the persistence of the reactive security strategy in contemporary practices can be attributed to several factors:

1. **Longevity**: The reactive model has been in existence for an extended period and encompasses elements that have been proven effective over time.

2. **Variety in Security Software**: The market is replete with a plethora of tools and resources designed for reactive security strategies. This diversity allows businesses to select solutions that align best with their specific needs, offering a degree of customization in their security approach.

3. **Simplicity and Cost-Effectiveness**: Inherently, the reactive approach is relatively straightforward to implement and maintain, often entailing lower ongoing costs compared to more proactive strategies.

However, it is imperative to recognize that the evolving nature of cyber threats necessitates a shift towards more dynamic and anticipatory security measures. This shift challenges the traditional reactive paradigm, advocating for an approach that is not only responsive but also preemptive in identifying and mitigating potential security risks.

On the other hand, proactive approaches such as prediction and forecasting can aid the network security state with different approaches. First, prediction can alleviate the risk to network security by predicting an attack based on the initial signs or the attack's ultimate goal. Second, forecasting can help with attack projection, where we track the attack steps to determine the attacker's next step. Consequently, we will be able to prevent additional harm or stop the attack in its early stages. Third, adopting security prediction and forecasting techniques such as situational awareness can increase the network's ability to sense and identify tampered messages. Finally, attack prediction and forecasting can help preemptively infer the attack itself or the attack's next steps.

REACTIVE	**ACTIVE**	**PROACTIVE**
Limited threat landscape view	Real-time monitoring	Full threat landscape overview
Plentiful resources	Limited resources	Difficult to achieve
Does not engage attackers	Engages attackers	Diverts or pre-identifies attackers

Table 2.1: Comparison of reactive, active, and proactive security strategies

Table 2.1 provides a succinct comparison of three distinct security strategies: reactive, active, and proactive. In the reactive approach, there is a limited view of

the threat landscape and a lack of engagement with attackers, albeit with an abundance of existing resources and options. Conversely, the active strategy emphasizes real-time monitoring and defense in depth, leading to increased security and direct engagement with attackers as they strike, though it is hampered by limited resources and the availability of skilled workers. The proactive approach offers an overview of the threat landscape through cyber intelligence feeds and real-time monitoring. It actively diverts attackers using techniques like honeypots and preemptively identifies potential attack vectors. However, this approach is challenging to implement and demands highly skilled personnel, who are often scarce.

2.3.1 Challenges and Limitations in Attack Prediction

Implementing predictive strategies in a practical context brings to the forefront two primary issues: accuracy and efficiency. While high accuracy rates are achievable in controlled dataset evaluations, these numbers do not necessarily translate to similar success in live network traffic, where prediction accuracies might score lower percentages. This discrepancy underscores the challenge of ensuring reliable attack prediction in dynamic, real-world environments.

Another significant consideration is the time required to respond to a predicted attack. The response time metric is crucial, as it determines the window available for preparation and defensive measures. This aspect of prediction efficiency, alongside the temporal proximity to the predicted event, is essential in evaluating the practicality of attack prediction methods.

Furthermore, two significant issues confront the implementation of attack prediction: the development of a knowledge base detailing potential attacks and the decision

of the optimal level of abstraction for attack prediction deployment. The latter is particularly pertinent, given the computational demands of implementing such systems, especially at the network level.

A pervasive problem in this field is the inadequacy of many existing datasets for evaluating attack prediction methods. Often, these datasets are not explicitly designed for attack prediction evaluation, leading to gaps in the applicability of their results. In real-life scenarios, evaluating attack predictions poses ethical and logistical challenges, as it is not feasible or responsible to allow adversaries to execute intrusions in actual networks solely for assessment purposes. Additionally, accessing every potentially compromised host in large network settings presents a formidable challenge.

In summary, it is insufficient to rely solely on accuracy when setting metrics for evaluating and comparing different attack prediction methods. A holistic approach must also consider the efficiency of the prediction and the critical factor of time remaining until the predicted event. This detailed evaluation framework is essential for developing effective and practical attack prediction strategies in the complex landscape of network security.

2.4 Dynamics of Proactive Security Measures

In this section, an extensive review of the literature on attack prediction and forecasting is conducted. This discussion encompasses security prediction and forecasting across various network architectures. It is important to note that while these strategies are discussed in a broader context, they hold potential applicability to VANETs

with appropriate modifications. This perspective allows for a comprehensive understanding of the methodologies and their adaptability to the specific nuances of VANETs environments.

2.4.1 Attack Dynamics: Proactive Security from Attack's Perspective

Given the diverse array of attacks in VANETs, each posing distinct challenges to network safety [28], this subsection undertakes a review of prior research from the standpoint of attack types. The categorization of existing studies is based on the nature of the attacks in VANETs. Our investigation into the field of attack prediction and forecasting revealed a disparity in the volume of research related to different attack types. For instance, attacks such as Sybil and denial of service (DoS) are extensively studied, whereas others like Forgery and Masquerading have received comparatively less attention.

Predicting Attack Rate

Statistical methods have been extensively employed in predicting attack rates. These approaches are useful for forecasting trends in the overall rate of cyberattacks [29] or specific attack patterns, such as variations in the usage of bots [30]. For example, Zhan et al. [31] utilized data from low-interaction honeypots and applied statistical analysis to predict attack rates with commendable accuracy, achieving predictions up to one hour in advance. Subsequently, Zhan et al. continued this research.

In recent years, non-statistical methods, particularly those involving machine learning and deep learning, have gained increasing traction. Compared to traditional statistical models like the Auto-Regressive Integrated Moving Average (ARIMA),

deep learning techniques have demonstrated superior performance in prediction accuracy [32], reducing error rates significantly by over 80%. Xing Fang et al. [33] employed bi-directional recurrent neural networks with LSTM in developing a framework to predict attack rates, finding that their neural network-based model outperformed statistical methods in terms of prediction accuracy. While the incorporation of deep learning and machine learning in attack rate prediction is still nascent, the initial findings indicate a promising avenue for future research in this area.

General Attack Prediction and Forecasting

The shift towards attack prediction and forecasting is increasingly recognized as a crucial development in the field of cybersecurity [34]. Despite its importance, research in this area is not as extensive as in attack detection. General attack prediction and forecasting encompass a broad spectrum, not limited to specific types of attacks. A fundamental aspect of attack prediction involves identifying potential targets or objectives, such as predicting vulnerable hosts and insider attacks. The work reviewed in this section does not focus on a specific attack.

In the context of predicting attacking hosts, Prasath et al. [35] employed machine learning techniques to identify and subsequently disconnect such hosts. They utilized a Bayesian algorithm model trained on historical network attack data, achieving an average prediction accuracy of approximately 91.68%, aligning with findings in [36].

Predicting insider attacks presents a formidable challenge, given the complexities involved. Azaria et al. [37] addressed this issue by focusing on stealth malware and countermeasure techniques. Their approach involved training a supervised machine learning model on mostly unlabeled and highly imbalanced data with minimal user

history. This scenario is akin to that in VANETs, where new users lacking an extensive historical record are commonplace. Despite these constraints, the algorithms tested yielded realistic and acceptable precision and recall outcomes. In the realm of connected vehicles, Haider al-Khateeb et al. [11] utilized a recursive Bayesian estimator, specifically the Kalman filter, for proactive anomaly detection. By constructing a dataset comprising event routes and GPS coordinates, they demonstrated the feasibility of proactively detecting deviations from normal routes, a significant advancement over traditional reactive solutions.

Specific Attack Prediction and Forecasting

Contrary to the broader scope of the general approaches discussed earlier, this section is dedicated to examining research initiatives that are specifically focused on particular types of attacks within VANETs. Given the extensive range of potential attacks in this domain, the analysis is selectively concentrated on a subset of the most prominent and perilous attacks that are known to pose significant threats in VANET environments. This targeted approach allows for a more in-depth exploration of strategies and solutions pertinent to these specific attack vectors.

Sybil: The prediction of Sybil attacks is particularly relevant in the context of Online Social Networks (OSNs), where users create Sybil accounts for malicious purposes. Al-Qurishi et al. [38] developed a deep-regression model aimed at predicting Sybil attacks within social networks. Their approach involved analyzing user profiles to extract characteristic features, which were then utilized to assess the trustworthiness of nodes and ascertain whether they were Sybil nodes.

GPS Spoofing and Position Faking: Dasgupta et al. [12] proposed a prediction-based strategy for detecting spoofing attacks. Employing an LSTM model, they analyzed a comprehensive real-world driving dataset, comma2k19 [39], containing diverse Autonomous Vehicle (AV) sensor data, such as Control Area Network (CAN), Global Navigation Satellite System (GNSS), and Inertial Measurement Unit (IMU) data. This data encompassed various parameters, including time, latitude, longitude, speed, steering angle, and acceleration. The model was used to predict the traveled distance for each timestamp, identifying discrepancies indicative of spoofing attacks. One specific attack scenario they examined involved the transmission of manipulated GNSS signals to create the illusion of an AV taking an exit while continuing forward. The LSTM model exhibited a high detection rate with minimal error and computational latency well within the latency constraints of VANETs.

Denial of Service (DoS): Lavrova et al. [40] focused on the prediction of DoS attacks utilizing the ARIMA model. They concentrated on prevalent DoS attacks, such as TCP flood, UDP flood, and ICMP flood, representing network traffic as time series data aggregated from network packet parameters. Indicators of attacks, such as an excessive number of SYN packets relative to ACK packets in TCP flood attacks, were identified. Parameters such as the count and average size of TCP packets were used to detect potential threats. However, this work did not provide specific metrics on the accuracy of the attack predictions, leaving room for further empirical validation.

2.4.2 Technique Dynamics: Proactive Security from Techniques' Perspective

This subsection delves into the varied methodologies employed in security prediction and forecasting within the domain of proactive security. The spectrum of attack prediction and forecasting methodologies is diverse, encompassing discrete models, continuous models, and notably, machine learning models [41]. Among these, Machine Learning (ML) approaches, particularly Neural Network (NN) and Supervised Learning (SL) models, have been predominantly utilized. Despite the prominence of ML techniques, the presence of other methodologies signifies the nascent stage of this field. Given that the research primarily focuses on ML-based approaches, this subsection will concentrate on studies related to machine learning in security prediction and forecasting. The ensuing discussion will cover various aspects of these studies, including the learning model employed, the datasets used, and the outcomes of the predictions.

Neural networks are renowned for their capacity to tackle complex problems, yield satisfactory results, handle incomplete data, and execute time-series predictions. In the realm of cybersecurity prediction and forecasting, neural networks have emerged as a leading technique. Fang et al. [33] developed a cyberattack rate prediction framework, bi-directional recurrent neural networks with long short-term memory (BRNN-LSTM), using LSTM networks. While statistical methods have traditionally been favored for predicting attack rates, BRNN-LSTM demonstrated that deep learning and neural network-based models could surpass conventional statistical models like ARIMA and GARCH in performance. The BRNN-LSTM model was rigorously evaluated based on five real-world datasets, showing its applicability without the need

for retraining in all but one case. LSTM was also employed in the prediction of spoofing attacks [12], utilizing the public real-world dataset comma2k19 [39] to predict the distance between two consecutive vehicles. This approach enabled the real-time detection of spoofed attacks based on an error threshold, which involved calculating the positioning error of the vehicle's GNSS device and comparing the vehicle's immediate and projected future locations.

Supervised learning has been a prominent approach due to its straightforward implementation and a wide range of classification and regression models. Among these, the Support Vector Machine (SVM) model has seen extensive application. Gao et al. [42] used SVM in their work towards detecting Sybil attacks in social networks. SVM was used to generate a probabilistic estimation for user trustworthiness and then compare the resulting score to a threshold that resulted in labeling the user as Sybil or benign. However, the use of SVM was limited, especially the prediction part, as the work focused on detecting, not predicting, Sybil attacks. Uwagbole et al. [43] used a Two-Class Support Vector Machine (TCSVM) to detect a Structured Query Language Injection Attack (SQLIA). This work introduced a dataset containing patterns extracted from SQL attacks. It involves the SQL attack's tokens and injection points to overcome the absence of such datasets. They trained an SVM model and tested its ability to predict the occurrence of an SQL attack by deploying it as a web service of a dot NET application named NETSQLIA. The model showed good results with 98% accuracy, but a predicted limitation in their work is that the model will need to be retrained in every new environment.

Following SVM, Bayesian networks emerged as the second most prevalent supervised learning technique for attack prediction. Okutan et al. [44] decided to use

non-conventional signals to predict cyber attacks in their work. Bayesian network classifier was fed with four different types of non-conventional signals (Twitter Attack Mentions (TAM), GDELT Event Mentions (GEM), GDELT Event Tone (GET), and Hackmageddon Number of Attacks (HNA)). The classifier was built to detect three attacks, DoS, defacement, and Malicious Email/URL (MEU), on the next day of the collected signals' timestamp. The model used these general signals to predict attacks on an anonymized company called KNOX, and it showed low but promising results in predicting each type of targeted attack. The model showed better results in a mixed experiment predicting any attack, not a specific one. However, the low accuracy rate (zero in the case of the DOS attack and 0.66 and 0.87 for the other two attacks) is still considered good since the signals were not related to the target entity and were collected based on different data sources.

The naive bayes model was also utilized effectively in cybersecurity prediction. Onoh used a different type of data and publicly available data to predict cyberattacks in his work [45]. The collected data were related to brute-force/password-guessing attacks. They used public data repositories (Twitter, Facebook, Instagram, blogs, and forms) related to the targeted attacks. Using a naive bayes classifier, the model could predict the attack with an F-measure of 70%. The results are still considered promising as only publicly available data from an external organization was used to predict the attack.

Moreover, the random forest model, as employed by M.Prasath et al. [35], was combined with the Binary Particle Swarm Optimization (BPSO) algorithm to predict and detect unauthorized users in the Software-Defined Networking (SDN). The KDDTeast21FS dataset was used as historical data to train the RF model, while

the binary particle swarm optimization algorithm was used for the feature selection process. The proposed scheme showed high prediction results for potential attack connections and potential attacked hosts with low detection sensitivity.

The reviewed cybersecurity prediction and forecasting work did not heavily utilize the unsupervised learning models as the supervised ones. Still, Al-Khateeb et al. [11] used recursive Bayesian and Kalman filters for behavioral analysis. They claimed that the used approach is more efficient than supervised machine learning as the latter heavily depends on a time-consuming training part, which will not be efficient for real-time prediction. Furthermore, the Kalman Filter does not need all the earlier system stages as it can only depend on the previous one. The model detected the deviations proactively from regular routes based on vehicle profiling. Accordingly, the model estimated future states.

2.5 Summary

This chapter has laid the groundwork by reviewing the current state of research in VANET security. Our review reveals a significant gap in proactive security strategies within VANETs. Notably, certain types of attacks have seen little to no research on proactive measures, which are focused on preventing attacks before they happen. The hesitation to adopt these strategies isn't because of the time-sensitive nature of VANETs. Instead, it arises from the difference in how advanced and straightforward reactive research is compared to proactive research, with reactive methods often being more developed and easier to measure.

A key reason for the gap previously mentioned is the unclear distinction between proactive and reactive security approaches in existing research. This ambiguity makes

it difficult to develop systems that are genuinely aimed at preventing security breaches before they occur rather than merely responding after they happen.

We have also observed that most reactive measures are implemented at the authentication level, which could give the impression that a proactive approach has been established, as it prevents malicious entities from infiltrating the system, ostensibly ensuring no future harm. However, since VANETs must contend with insider attackers or misbehaviors due to the open sharing environment, there is a clear need for proactive security approaches. These approaches should specifically target the interactions between pre-authorized network entities or those who have passed the authentication step but might choose to act selfishly at any time.

In response, our research adopts a clear stance on proactive security within the TMS for VANETs. In the upcoming chapters, we will apply proactive security principles across various components of the TMS, clearly defining what makes an approach proactive versus reactive. This strategy not only aims to strengthen VANET security against emerging threats but also establishes a clear framework for future research. By outlining a clear method that distinguishes between anticipatory and responsive security tactics, we pave the way for future studies to integrate proactive measures into their security frameworks more easily, contributing to the advancement of vehicle network security.

Chapter 3

Situational Awareness for Misbehavior Detection

In this chapter, we exploit the role of Situational Awareness (SA) in enhancing proactive security measures within network environments. Situational awareness aids in forecasting potential threats by leveraging historical data and integrating it with current trends to predict future network vulnerabilities. Specifically, we investigate the application of situational awareness to the trust subject component of the TMS. Since the trust subject in a TMS refers to the entity whose trustworthiness is being evaluated, effective situational awareness enables the TMS to preemptively prevent interactions with anticipated malicious entities within the network, thus significantly strengthening the network's overall defense.

3.1 Introduction

As a specialized form of ad-hoc networks, VANETs inherit the complex security issues of these networks, making security a critical concern. [5, 6]. VANET messages are broadcast across the network, enabling all participants to share road conditions and personal data. Hence, VANET nodes must distinguish between malicious and benign messages [26]. Message authenticity and correctness are the main factors for achieving

a secure and safe communication process. Message authentication has been the focus of much research, culminating in the issuance of several standard protocols (IEEE 1609). However, ensuring the accuracy of messages remains a challenge.

BSMs are one of the most critical exchanged messages in VANETs. Typically, BSMs are transmitted at a rate of 10 messages per second, although this frequency can vary depending on the vehicle's speed or the specific requirements of the traffic system. BSMs are periodically broadcasted by vehicles to share critical information such as position, speed, heading, and brake status with nearby vehicles and infrastructure. Such information is used for various purposes, such as collision avoidance and traffic jam detection. Unfortunately, due to the openly accessible and unencrypted format of BSMs, malicious nodes can broadcast false information, resulting in devastating consequences. [23].

The open sharing environment in VANETs leads to a vulnerability to a specific type of attack called misbehavior attacks, where vehicles share misleading information. Misbehavior messages are authentic messages that contain false information [27], and exchanging such erroneous information can propagate quickly. Additionally, vehicles and driver profiling may depend heavily on this information. Thus, assessing the trustworthiness of the surrounding nodes is crucial because we need to ensure the accuracy of the received information and safeguard the driver's decision-making process from being influenced by false information.

Content integrity requires critical and integral decision-making. However, traditional network security mechanisms trust whoever gets inside the network. Hence, conventional all-or-none security mechanisms cannot deal with attacks lacking message correctness. Trust evaluation has established itself as an asset for enabling

secure and reliable exchange of information between network entities [46]. Accordingly, utilizing ML for attack classification and trust evaluation for inside attacks (e.g., misbehavior attacks) can lead to a more secure environment in VANETs.

Situational Awareness (SA) has shown promising results in the field of cyber defense [47]. SA is about reaching the state of being aware of the circumstances around us. Primarily, we are interested in the events that are particularly relevant to the current situation. Ultimately, deploying SA in VANETs for security depends on the environment and the behavior of the surrounding vehicles. Furthermore, SA relies on the amount and quality of the provided data. Therefore, the more information provided, the better the awareness of the surroundings.

This chapter demonstrates using SA on ML models' results to detect misbehavior attacks. In this context, we quantify situational awareness in VANETs by calculating a trust score using collected information from the environment. Subsequently, the trust information is fed to the situational awareness model. The result is a misbehavior detection system that uses the powerful classification of ML and augments the accuracy of these results by utilizing the trust information obtained from the situational awareness model. The major contributions presented in this chapter can be summarized as follows:

- introduce Situational Awareness with Machine Learning for Misbehavior Detection (SAMM): A novel approach that enhances misbehavior detection by integrating situational awareness with machine learning.

- evaluate the approach on a publicly available dataset and different types of machine learning models and demonstrate better results than those ones obtained with just the machine learning models.

- contribute toward standardizing the VeReMi dataset by providing new baselines for future researchers.

The rest of this chapter is organized as follows. Section 3.2 discusses the work related to misbehavior detection, machine learning, and situational awareness in VANETs. In Section 3.3, we discuss the attack model along with the dataset and situational awareness. Section 3.4 provides a detailed description of SAMM, the misbehavior detection model using situational awareness. Section 3.5 highlights the simulation results of the proposed model. Finally, Section 3.6 concludes the chapter.

3.2 Related Work

3.2.1 Misbehaving Detection

Misbehavior detection focuses on identifying messages with incorrect or anomalous data. Misbehavior detection can be divided into node-centric or data-centric [48]. Node-centric considers the data correct if the sender can be trusted [49]. Data-centric verifies the data to determine its correctness. We outline some of the existing work that discusses misbehavior detection. Bissmeyer et al. [50] proposed a method where message correctness verification depends on central decision-making authority. A central place receives the message data and decides if the message is truthful or misbehaving. This approach suffers from dealing with a voluminous amount of data. The work by Grover et al. [51] can be considered as an example of a consensus method. In this scheme, a group of nodes tries to identify malicious nodes that perform misbehaving. Abu-Elkheir et al. [52] proposed a position verification scheme to help vehicles to be more acumen for other vehicles' announced positions. They used direct and 2-hop neighbors to create a lower and upper boundary for the sender

position, and if the sent position is beyond this plausibility area, a flag will be raised. In the absence of 2-hop neighbors, they used the Received Signal Strength Indicator (RSSI) to verify announced distance against relative distance to distance measures. The proposed scheme showed promising results for position verification in VANETs.

Deploying a Misbehavior Detection System (MDS) to identify internal attacks has shown promising results. Different works [53, 54] adopted ML-based MDS to detect falsification attacks. Khattab et al. [55] proposed an IDS which extracts features from trace files to detect attacks in VANETs. They used an Artificial Neural Network (ANN) and fuzzified data to detect abnormal behavior of vehicles.

So et al. [56] produced ML-friendly features using the VeReMi dataset [16]. They compared the effectiveness of relying only on plausibility checks against using plausibility checks as amalgamated features within the ML models. Finally, they evaluated the model's ability to classify various attack types. In addition to adopting ML in MDS, measuring trustworthiness demonstrated how powerful an ML tool could be for misbehavior detection, as in [57].

3.2.2 Situational Awareness

Many domains utilize Situational Awareness (SA) (e.g., robotics, psychology, AI, and computer science) [41]. However, utilizing SA for VANET security is still in its early stages. Hong et al. [58] proposed a Situational Awareness Trust Architecture (SAT). SAT addressed three aspects (trust attributes or policy control, proactive trust, and social trust). In the proposed architecture, trust attributes consist of entity trust and data trust. Also, they showed how to set up the trust in advance by using the vehicular network instead of waiting for a particular event to occur. Finally, if there is no RSU

infrastructure, they suggested using social networks to gather surrounding vehicle assessments. Thus, this work proposes utilizing situational awareness for establishing trust in VANETs, but it lacks real or simulated experiments.

The work in [59] investigated the network security situation as a nonlinear time series and used a Radial Basis Function (RBF) neural network as it can achieve a nonlinear combination between the input space and the output space. In this case, the input is previously stored values about the history and current situation, while the output is the following state situation with different n time windows. The time windows represent the selected number of continuous situation values (e.g., setting 3, 4, or 5 continuous situation values). The simulation experiment showed how the RBF neural network achieved better results than the backpropagation neural network in predicting the network security situation. However, an optimization algorithm can be used, or the RBF neural network structure may be adjusted to improve further.

Hashem et al. [60] used SA in VANETs to improve traffic safety. By utilizing the exchanged information through communication channels like BSMs in VANETs, they can be aware of the state of the network around the hour. In particular, they utilized SA to maintain the stability of the routing links and improve the quality of service (QoS). The routing algorithm updates the calculated QoS metrics if one established route becomes ineffective.

3.3 Preliminaries

3.3.1 Dataset and Attack Model

The VeReMi dataset [16] is publicly available and actively maintained on GitHub, ensuring its integrity. There are five different attacks represented in VeReMi: constant position, constant offset, random position, random offset, and eventual stop. VeReMi contains 225 simulations with three different traffic densities (low, medium, and high). Every vehicle that receives a message has a log file created, which consists of the sender vehicle ID, a unique message ID, position (x,y,x), RSSI, and velocity (x,y,z). Also, Mitchel et al. [61] discussed the need for a standardized dataset so that future researchers can validate their work. This motivation was behind producing the VeReMi dataset in the first place. So et al. [56, 62] used a modified version of VeReMi to create an ML-friendly metrics model, also used in one of our experiments, as will be discussed in more detail.

3.3.2 Plausibility Checks and ML for Detecting Misbehavior

One of the main approaches used recently for message correctness —and showing promising results— is the adoption of plausibility checks. Plausibility checks are used as a feature vector to enhance the ML model's classification capabilities [56]. In this experiment, we used several features and plausibility checks. Most were introduced in previous works. The used plausibility checks are as follows: 1) Displacement in X and Y positions, 2) Difference in X and Y velocities, 3) Acceleration in X and Y, 4) Predicted position in X and Y, 5) Predicted velocity in X and Y, 6) Difference between calculated and predicted position in X and Y, 7) Difference between calculated and predicted velocity in X and Y, 8) Distance between X and Y, and distance

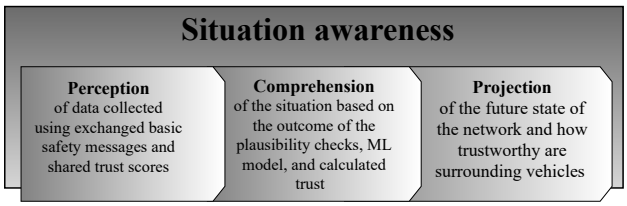


Figure 3.1: Situation awareness phases in the VANET trust model

between predicted X and Y.

3.3.3 Situational Awareness

Situational Awareness (SA), in its basic form, is about knowing what is happening in your vicinity. Figure 3.1 shows the three phases of SA and its adoption in the context of VANET security for detecting misbehavior attacks. In the first perception phase, we rely on the data contained in BSMs. Then, in the second step of comprehension, we use the collected data to calculate plausibility checks, feed them to the ML model, and provide it to the trust equation. Finally, the projection step considers the ML and the trust model scores to assess the situation at that particular moment and predict the imminent situation's development.

One major challenge in SA is how to measure it. Quantitative analysis is used to represent the state of the network security situation at one point in time [41]. While quantitative analysis can provide warnings about the network security situation, it suffers from the lack of information regarding the nature of the attack. Nevertheless, some well-known techniques for measuring situational awareness are inherited from military aviation testing, like subjective ratings, simulation freeze, and task performance measures. In our case, we are extending one of these measuring techniques-Subjective Ranking Technique (SART)- to apply it in the security domain. In the

subjective ranking, a driver's ranking is based on ranking by observers or ranking by other drivers. In this work, we measure situational awareness by assigning a trust score to the vehicle based on other drivers' reported observations.

3.3.4 Trust in VANETs

It is essential to ensure the integrity of exchanged data in VANETs, particularly BSM data. Trust management systems in VANETs use related information as values for trust functions or as inputs for the trust inference models. The trust score or the inference results can determine the trust level of a certain vehicle. Accordingly, the model used the collected data from BSMs and data from the environment as values for the trust calculation function, Logistic Trust (LT) [63].

Logistic Trust Model

LT was adopted for measuring SA in terms of how trustworthy the vehicles in the network are [63]. LT is very compatible with SA as it is time and experience-driven. SA tends to collect data iteratively over time, contributing to the LT trust scoring process. The logistic trust model consists of three parameters: dissatisfaction, flag, and expectation.

Dissatisfaction: This measures the contradiction of the received information from vehicle s. If $i_n(s)$ is the number of incorrect messages s sent and $v_n(s)$ is the total number of sent messages by s, then dissatisfaction in time-slot n is

$$\delta_n(R, s) = \frac{i_n(s)}{v_n(s)} \tag{3.1}$$

while the total dissatisfaction up to time slot n is

$$\bar{\delta}_n(R,s) = \frac{\sum_{n=1}^{n} \delta_n(R,s)}{n} \tag{3.2}$$

Flag: In LT, the flag parameter is part of the direct trust. Also, if the dissatisfaction score was higher than a certain pre-declared threshold β, a flag $\alpha_n(R,s)$ is raised for this s vehicle. We modified this part by making the flag part of the indirect trust and raising the flag after a certain number of attacks. We did this because LT was designed for ad-hoc networks in general; this made the flag parameter more tolerant of misbehaving. Initially, the flag was lowered if the dissatisfaction score went below the threshold after more interactions. In VANETs, vehicles have limited interaction time, leading to fluctuating dissatisfaction scores. Therefore, we found that vehicles should be flagged after a certain number of attacks. If a flag was raised by error, the overall trust score does not solely depend on the flag state. Other parameters (e.g., expectation, dissatisfaction) may result in accepting the messages received from that vehicle.

Expectation: $\rho_n(R,s)$ is the mean of the recommendations regarding vehicle s in time slot n

$$\rho_n^{LT}(R,s) = \frac{b_n(s)+1}{b_n(s)+g_n(s)+2} \tag{3.3}$$

where bad recommendations are represented as $b_n(s)$, and good recommendations are represented as $g_n(s)$. Altogether, the final trust score is given by

$$t_n(R,s) = \frac{1}{1+e^{-}(\vec{V}.\vec{W}+W_0)} \tag{3.4}$$

where

$$\vec{V} = \{\bar{\delta}_n(R,s), \rho_n^{LT}(R,s), \alpha_n(R,s), t_{n-1}(R,s)\}$$

In equation 3.4, $\vec{W}$ represents a vector of weights that correspond to the elements in vector $\vec{V}$. Each weight in $\vec{W}$ determines the importance or influence of the respective element in $\vec{V}$ on the overall trust calculation. Additionally, W_0 is a bias term that adjusts the initial trust level for node n, denoted as $t_n(0)$, ensuring an appropriate baseline trust value at the starting point.

In our version of LT, the $\vec{V}$ is a vector of the values of the four main components of the trust equation: dissatisfaction, flag, expectations, and the trust score for the vehicle s in the previous time slot.

From the above discussion, we can divide the problem into two parts. First, we measure SA in terms of a trust score based on LT. Second, we integrate ML with SA. The first part will help utilize SA with ML and find a way to incorporate both in the decision-making process. The second part will lead to overcoming ML limitations and improving its detection rate.

3.4 Proposed Scheme

In this section, we describe the proposed situational awareness and machine learning scheme for misbehavior detection. The proposed scheme has two goals: measuring situational awareness to reflect the state of the network in a given time using logistic trust and adopting this holistic view with the acquired information from the used machine learning model to achieve better results in detecting misbehavior attacks.

Algorithm 3.1 SAMM algorithm (Part 1)

Inputs:
$Message_n$: Current received message Car_n: Sender vehicle
$\overrightarrow{A}$: Weight vector for the trust vector variables
$attackProb$: The ML model probability that this message is an attack message
$nonAttackProb$: The ML model probability that this message is a non-attack message
$dissatisfactionValue$: The dissatisfaction value
$dissatisfactionThreshold$: The dissatisfaction threshold
$attackThreshold$: The threshold of the message being an attack according to the user
$notAttackThreshold$: The threshold of the message being a non-attack according to the user
$trustAttackThreshold$: The threshold of the trusting score for not accepting the message
$trustnotAttackThreshold$: The threshold of the trusting score for accepting the message
$flagThreshold$: The threshold for raising the flag
Outputs:
$Misbehaving_n$: Deciding that the message n is a misbehaving message
$Benign_n$: Deciding that the message n is a benign message
Variables:
$trustScore_n$: Trust score in n
$flag_n$: A flag which indicates whether the sender is an attacker or not

1: **procedure** REVALUATING THE RECEIVED MESSAGE
2: **if** $Message_i$ is detected as an attack and Car_n is not in $attackerList$ **then**
3: add Car_n is in $attackerList$ and attack counter initialized by 1
4: **end if**
5: /*Calculate the trust score using the modified logistic trust model*/
6: **if** $Message_n$ is detected as an attack message by the ML model **then**
7: **if** Car_n number of attacks > $flagthreshold$ **then**
8: set $flag_n$ to 1
9: **end if**
10: $attackMessagesPercentage$ = ($receivedAttackesMessages$ / $receivedMessages$) * 100
11:

Algorithm 3.2 SAMM algorithm (Part 2)

12: **if** $attackMessagesPercentage > dissatisfactionThreshold$ **then**
13: set $dissatisfactionValue$ to 1
14: **else**
15: set $dissatisfactionValue$ to 0
16: **end if**
17: $expectationValue = (receivedAttackesMessages + 1) / (receivedAttackesMessages + receivednonAttackesMessages + 2)$
18: $trustScore_n = 1/1 + e^{\vec{V}.\vec{W}+W_0}$
19: **if** $trustScore_n < trustAttackThreshold$ and $nonAttackProb > notAttackThreshold$ **then**
20: $Benign_n$
21: **else**
22: **if** $Message_i$ detected as a non-attack and Car_n is not in $nonAttackerList$ **then**
23: add Car_n in $nonAttackerList$ and initialize a non-attack counter by 1
24: **else**
25: increment the non-attacks counter by 1
26: **end if**
27: **if** $trustScore_n < trustnotAttackThreshold$ and $attackProb > attackThreshold$ **then**
28: $Misbehaving_n$
29: **end if**
30: **end if**

3.4.1 Measuring SA Using LT

In order to measure SA, we utilized the SART concept [64]. In SART, observers in the network can rate a driver. Following the same concept in VANETs, other drivers can act as observers to rank a certain vehicle. LT [63], was used to rank vehicles by calculating their trust score.

In SAMM, when vehicle R receives a message from vehicle s, LT calculates the trust score for vehicle s using two sources of information: vehicle R's direct experience with vehicle s and indirect experience between other vehicles and vehicle s. LT uses

three parameters for trust calculations: dissatisfaction, flag, and expectation. In Section 3.3, we explained how each parameter is calculated. In this section, we will present how we adopted these parameters in our system.

Dissatisfaction

$\delta_n(R, s)$, measures the inconsistency of the received information from a certain vehicle. We measure dissatisfaction by calculating the difference between the correct and incorrect messages received from vehicle s. Initially, we create a counter for correct and incorrect messages for every vehicle. Then, whenever we mark a message as benign, we increase the correct message counter by one and vice versa. After that, we use Equation 3.1 to calculate the total dissatisfaction in the current time slot.

Flag

$\alpha_n(R, s)$, which in the system is related to how many attacks were detected from a certain vehicle s. A flag is raised if a vehicle exceeds a certain threshold in terms of the number of attacks.

Expectation

$\rho_n(R, s)$, reflects what other vehicles in the network think about vehicle s. We look at reported attacks from other vehicles as bad recommendations $b_n(s)$ and interactions without attacks as good recommendations $g_n(s)$. Consequently, we calculate expectation according to Equation 3.3

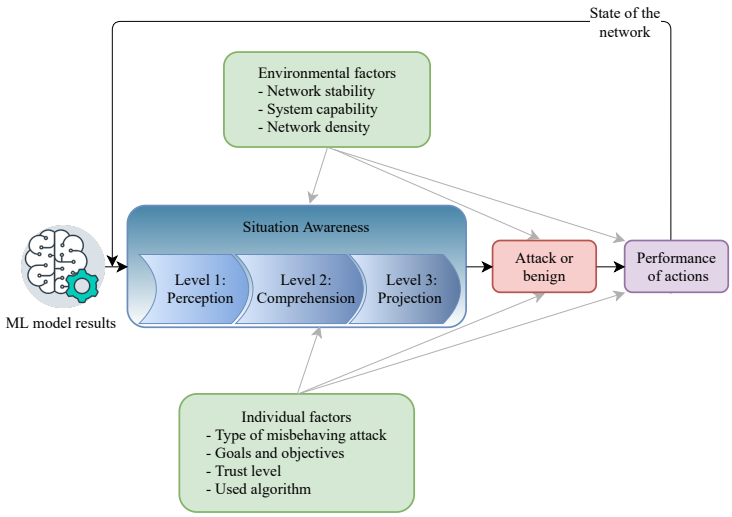


Figure 3.2: Situation awareness and machine learning model for detecting misbehavior attacks in VANETs

3.4.2 Integrating SA with ML for Misbehaving Detection

In this section, we discuss the main components of the SA model as shown in Figure 3.2.

System Input

Generally, in SA systems, the input is the state of the environment. This SA model was built to enhance the detection rate of ML models, and the input here is the output of ML models. Another difference is the way we build the used ML models. Plausibility checks are mandatory for the ML building process to build the ML model in a way that is related to the environment. Plausibility checks use the collected information from the environment, which is crucial for SA. We also did not take the ML output in terms of 0 or 1 but in terms of probabilities. Thus, it gives us a better understanding of how the system assisted the situation. Another input here is *feedback* and its role in enhancing the results in the future. We build the current

decision in time n, considering the previously collected and computed information in the time slot $n-1$. Also, in the LT algorithm, we include the previous trust score in time $n-1$ in calculating the trust score in time n.

SA Level 1 - Perception

Perception starts with collecting the surrounding vehicles' data stored in the BSMs to detect misbehavior attacks better. Such data include the vehicle's location, speed, and send time. Another source of information is the outcome of machine learning. The outcome is the probability that the received message is an attack. The general rule in SA is that more information leads to better predictions.

SA Level 2 - Comprehension

Misbehavior detection needs to use the provided elements to go beyond being aware of them. Thus, analyzing and using the information provided is a must. For instance, we calculate the trust elements of the trust function, like the ratio between false and benign messages or other vehicles' recommendations about a particular vehicle, to compute its reputation.

SA Level 3 - Projection

The projection step concerning misbehavior detection leads to the capability of estimating other vehicles' levels of trust. Accordingly, this is achieved through building knowledge of vehicles' behavior and the awareness of the current situation (outcomes of SA level 1 and level 2). In level 3, we are trying to determine the most trustworthy

vehicles and vice versa. Also, we are looking for vehicles that showed signs of misbehaving before and see if they exceeded the threshold of misbehaving. Ultimately, this will be used in the next step to enable the most suitable decision at the current time. Hence, it determines whether to accept the received message or not.

Environmental Factors

Various elements like network stability and density affect the situational awareness process. Additionally, deploying sensors and security software plays a role in the system's environment. System factors assist different levels of SA and cooperate in the decision-making process. In our case, we assume that network stability and density are fixed and the same for all vehicles.

Individual Factors

Individual or group attributes influence the SA process [65]. Individual goals and objectives may change according to the environment, which leads to new plans to meet these new goals. In our case, the system operator may change the trust acceptance threshold for a better detection rate; also, it can have a different algorithm to call up according to the situation. Our experiments include a set of location forgery attacks for an algorithm.

3.4.3 SA and ML Algorithm for Detecting Misbehavior Attacks

As shown in Algorithms 3.1 and 3.2, after receiving a classification result regarding a recently received message from a surrounding vehicle (line 6), we run the algorithm to either confirm this classification or change it based on the SA decision. The algorithm

makes use of the data collected from the environment by calculating dissatisfaction (lines 13 and 15), expectation (line 17), and raising flags (line 8) according to the recent situation. Finally, we combine two main sources of information: the ML model decision and the calculated information based on the environment to reach a final decision. ML model information comes in the form of probability, not discrete decisions. Therefore, instead of saying this is an attack message or not, we are more interested in the probability of this message being an attack or a benign one. Furthermore, the environmental information reflects SA and predicts the value of the surrounding vehicles' trust. Accordingly, we use these two values (classification probability and trust value) to keep the initial classification obtained from the ML model regarding the received message or change it as shown in the *if* statements before line 20 and line 28. Finally, we accept a message classified as an attack or vice versa (lines 20 and 28).

The algorithm's complexity is $O(n)$. Even though we search multiple lists (e.g., attacker list, non-attacker list), we still do not have any nested loops. If we consider n, m, which are the lengths of these lists, this makes the big O equal to n because $O(n+m+...)$ is equal to $O(n)$. Accordingly, integrating SA should not pose a consider overhead with an adequate big O.

3.5 Performance Evaluation

After utilizing SA, we conducted four experiments to understand the improved aspects and their nature. The first experiment (section 3.5.1) differs from the other three and can be considered a pre-experiment. Nevertheless, we included it in this section as it gives an idea about the effectiveness of the SA algorithm.

We conducted the other three experiments on ML models to demonstrate the influence of SA on the ML detection rate for misbehavior attacks. We used a publicly available dataset named VeReMi for these experiments to test a wide range of misbehaving attacks. We established the ground truth in the second experiment (Section 3.5.2). Therefore, we built the ML model and calculated the plausibility checks in make SA the only variable in the equation. Thus, we compared the detection results between the ML model and the ML model with SA to see the impact SA had on the accuracy rate of detecting misbehavior attacks. The third experiment (Section 3.5.3) helped us refine the understanding of the effect of SA on identifying misbehaving. Furthermore, we compared our work against a recent work that used federated learning on top ML instead of SA, as in our case. Finally, in the fourth experiment (Section 3.5.4), we further demonstrated the ability of SA to enhance ML models' capabilities in detecting misbehavior attacks.

3.5.1 Correlation Between SA Calculated Trust and Attack Density

This experiment was conducted to know the performance of the SA measuring trust algorithm. This experiment aimed to see the correlation between the measured trust expectations for the vehicles in the network and their actual behavior over time. Therefore, we have the calculated trust value (inverted for better presentation), and we have the reported attacks based on ML detection.

We used a scatterplot to represent the correlation between the **inverted trust** value and the **number of detected attacks** for each vehicle. Figure 3.3 shows the correlation in all five attacks and the correlation coefficients at the top left corner. Each dot in the plot reflects a vehicle in the scenario. The dot's location depends on

the number of detected attacks for this vehicle and the calculated trust score by the algorithm. The error region is also illustrated in light blue in Figure 3.3. This error region corresponds to the 95% confidence interval of the estimate.

Positively correlated variables move in the same direction, as shown in the graph. We used the inverse of the trust value to simplify the scatterplot's visualization between calculated trust and detected attacks. Also, the correlation coefficient on the top of the graph reflects the association between two variables. Variables with a correlation coefficient magnitude over 0.5 are considered moderately correlated. The minimum obtained correlation was 0.59, and we reached 0.71 in the random offset attack. The scatterplots for all the attacks show a small gap between the trust scores. The gap is not big, but it resulted from the outliers nodes (i.e., vehicles with a small number of attacks with high trust scores and vehicles with a large number of attacks with low trust scores). We added an area around the nodes (the small blue circle) related to this node's number of detected attacks, and a color bar was used to show where most nodes lay and how small the outliers' numbers are. The gap may look smaller in the first three attacks, but the scale is too small. Accordingly, "small" here does not reflect more density because the difference is insignificant. This shows how the algorithm's trust results correlate to the vehicles' behavior in the actual scenario. However, this experiment was conducted as a proof of concept for our assumption. Thus, we cannot interpret the result more coherently as to how the correlation coefficients and the types of attacks are related. However, this will be done in the subsequent experiments.

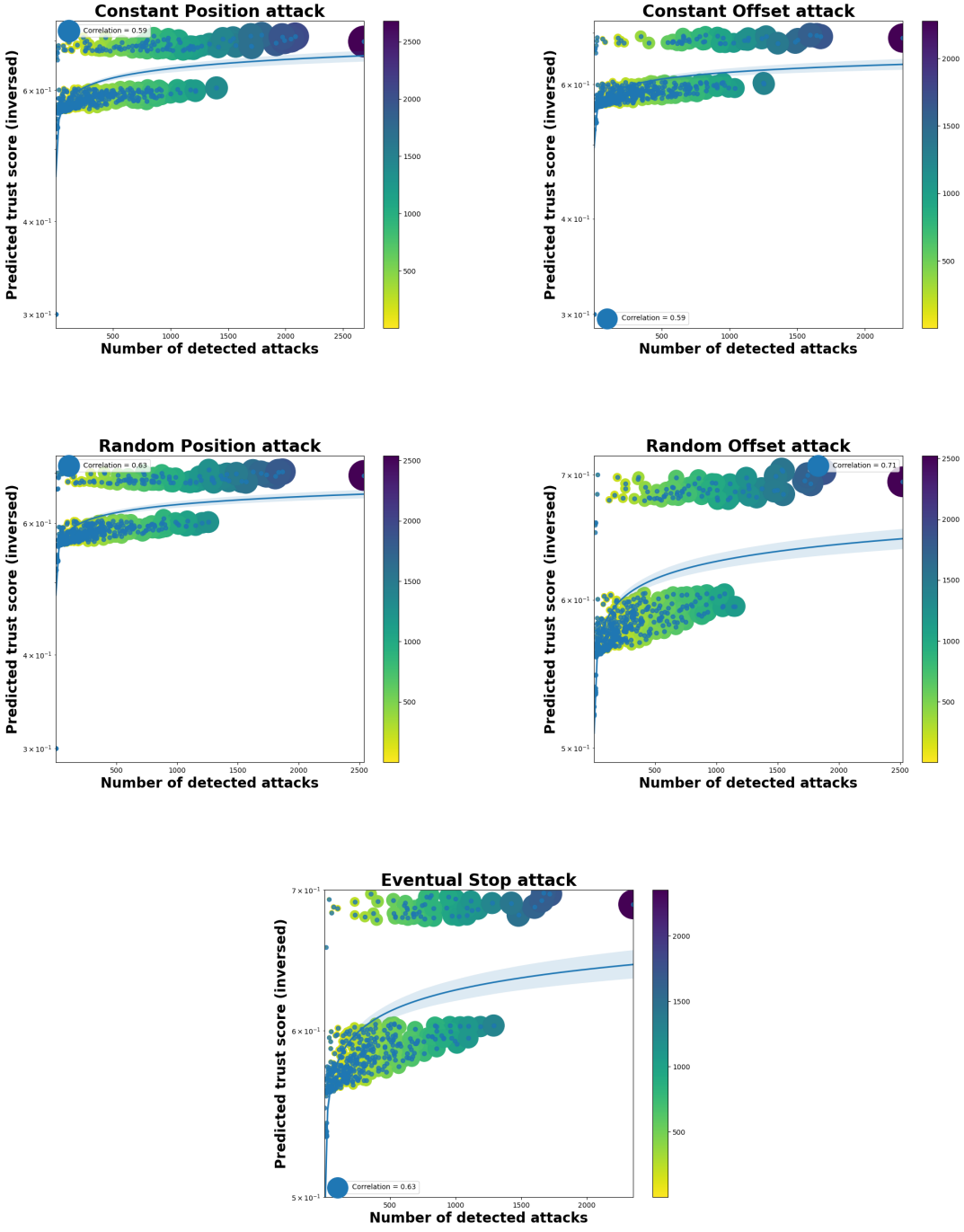


Figure 3.3: Correlations between SA trust and attacks density

3.5.2 Analyzing the Effect of Using SA on ML Detection Rate

To fathom the merging between ML and SA, we needed to investigate the outcome of the plain ML model against itself after synthesizing SA. For this experiment, we designed the ML model and the plausibility checks. Consequently, we made SA the only variable in the equation.

Parsing the Dataset

We started with importing all VeMeMi `.json` files. Each file belongs to one vehicle and contains the received BSMs from its neighbors in a 300-meter range. Also, attacks files are categorized into three different density scenarios low (35 - 39 vehicles), medium (97 - 108 vehicles), and high (491 - 519 vehicles).

Because SA reflects the system situation at a specific time, we sorted all the records in chronological order. Most importantly, this differs from most of the approaches used, which mainly focus on pairing the sender and the receiver messages. This pairing process results in changing the chronological order of the records. Of course, the traditional approach is not wrong, but it only focuses on creating attack features without considering time as one of the elements. After that, we converted all the files to `.csv` file format without concatenating them together.

Plausibility Checks and Supporting Features

We calculated eight different plausibility checks and features, which we represent in Section 3.3. Some features were calculated as plausibility checks, while others were calculated to gain more insight into the environment. Subsequently, feature numbers 1, 6, and 8 are the plausibility checks, while the others act as supporting features.

We needed the supporting features to expand the understanding of the behavior of the surrounding vehicles and for calculating plausibility checks (e.g., calculating acceleration a to calculate position later).

Random Forest and Trust Calculation

Many studies have adopted machine learning models for anomaly detection [66], with the random forest classifier demonstrating remarkable accuracy, particularly when handling extensive data.

The first and second experiments were implemented in Python 3. We used the scikit-learn library (`sklearn.ensemble`) for the random forest classifier. We divided the data into 0.75 for training and the rest for testing. The number of used trees is 100. After training the model, we assessed and saved the accuracy score for each of the five attacks. Afterward, we ran the obtained output through Algorithm 3.1 to observe the effect of including SA on the final results.

Table 3.1 demonstrates the difference in performance between the results obtained from the ML models and the results after including SA in all three traffic density scenarios. We observe a significant increase in the total accuracy rate on the various types of attacks in different densities, except for the first attack in the high density. The reason behind this is mainly because of the simplicity of this attack. Using displacement as a plausibility check for this attack makes detection accurate near 100%, and any small change would significantly affect the result. Additionally, SA considers outside elements, not just the plausibility checks, which affect the results.

Even though using SA has demonstrated its ability to enhance detection accuracy, it showed a limitation in the near-perfect detected attacks. Utilizing SA will be good

with low or imperfect detected rate attacks; otherwise, the extra information and insights will not add to the prediction quality as with the first attack earlier. Another reason behind such behavior is the type of injected information. We are not filtering the information based on the targeted attack to make the framework as generic as possible and ensure that the data will not be fixed to what we are trying to detect. Still, more than 90% of the cases show a very promising increase in the detection rate.

Traffic density	Low		Medium		High	
	ML	ML & SA	ML	ML & SA	ML	ML & SA
Constant position	0.982	0.983	0.970	0.980	0.999	0.612
Constant offset	0.859	0.878	0.807	0.837	0.830	0.860
Random position	0.849	0.861	0.880	0.890	0.910	0.920
Random offset	0.843	0.861	0.813	0.842	0.870	0.890
Eventual stop	0.867	0.906	0.904	0.922	0.910	0.920

Table 3.1: Comparison between ML detection results with/without SA in different traffic densities

3.5.3 Examining the Effect of Using SA on Recall and Precision in ML

In this experiment, we compare our approach to attack detection in VANETs, which leverages Situational Awareness (SA) with machine learning, against the method in [67]. While traditional research in misbehavior detection often focuses on enhancing ML models or feature detection, we integrate SA to improve detection accuracy. In contrast, the approach in [67] enhances ML models for privacy-preserving misbehavior detection by using Federated Learning (FL), which prevents sharing Basic Safety Message (BSM) data with third parties, resulting in improved precision, recall, and performance. Their research examines the benefits of FL over traditional ML models,

whereas this study explores the advantages of incorporating SA. Our experimental methodology aligns with theirs by comparing a basic ML model against an enhanced version, providing a direct assessment of the added value of the proposed approach.

We opted to use the same dataset, Vehicular Reference Misbehavior (VeReMi), and implemented the neural network in [67] given the published aspects. We constructed two sub-experiments. "In the first sub-experiment, we included plausibility checks in our SA-based model, as they are a core component of our proposed solution. In contrast, the FL model did not incorporate these checks. For the second sub-experiment, we applied the same plausibility checks to both the SA and FL models, keeping all other conditions constant. This setup allowed us to directly assess the impact of SA on detection performance, independent of the plausibility checks, and compare it with the FL approach under identical conditions.

In terms of evaluation, we followed their precision and recall evaluation matrix, as shown in Table 3.2. This will give us more insight into the behavior of SA on ML results. Given that precision and recall affect each other, we found out that SA enhances recall even if it is the cost of precision. For Attacks 1, 3, and 4 in Table 3.2(a), we can see a notable increase in recall in return for the small sacrifice in precision. Even for the second experiment, we can see a boost in recall in all the different attacks, as shown in Table 3.2(b) at the expense of a minor reduction in precision.

3.5.4 Improvement in the Recall by Utilizing SA

Next, we evaluate the model to improve the output results obtained by So et al. [56]. Their work increased the detection of misbehavior using ML and plausibility checks.

Precision					
	Constant pos.	Constant offset	Random pos.	Random offset	Eventual stop
FL Model	0%	-1%	0%	1%	-1%
FL & SA	-2%	9%	1%	-4%	9%
Recall					
FL Model	-13%	-4%	0%	0%	-16%
FL & SA	4%	0%	8%	15%	0%

((a)) The enhancement in terms of precision and recall by the federated learning model and our model (sub-experiment #1)

Recall					
	Constant pos.	Constant offset	Random pos.	Random offset	Eventual stop
FL Model	-13%	-4%	0%	0%	-16%
FL & SA	3%	11%	8%	7%	10%

((b)) The enhancement in terms of recall by the federated learning model and our model where same plausibility checks and features were used (sub-experiment #2)

Table 3.2: Examining the effect of using Situation Awareness (SA) on recall and precision in ML

We also compare the final results of their approach to ours. The final results in this chapter are represented in terms of precision and recall, following the same evaluation approach. Unlike the previous experiments, the used dataset is a modified version of VeReMi called ML-friendly VeReMi [56]. In this version, all the files of one scenario were concatenated into one `.csv` file. Moreover, two attributes were added to each `.csv` file named "Attacker Type" and "Receiver ID". The "Attacker Type" attribute contains a number from 0 to 5, where 0 indicates that this is a normal vehicle, and the rest of the numbers reflect one of the five attacks in the dataset. The mapping between the number and the attack type was obtained from the ground truth file, represented in the original dataset. Moreover, "Receiver ID" was added to build a

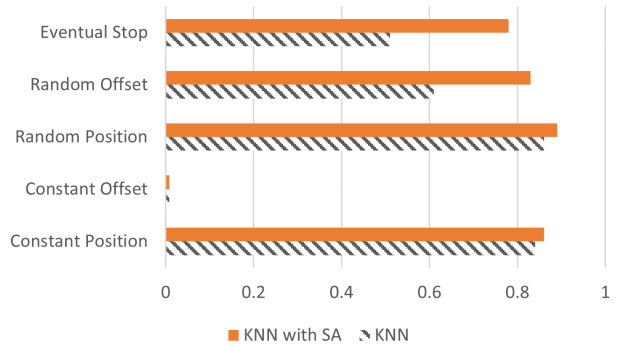


Figure 3.4: Recall comparison between the obtained results by the KNN model and after integrating SA

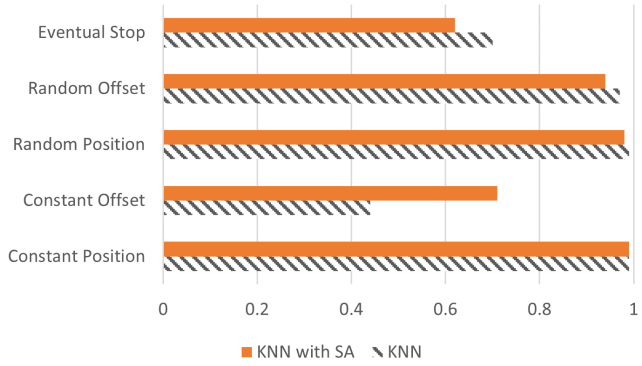


Figure 3.5: Precision comparison between the obtained results by the KNN model and after integrating SA

sender-receiver pair that helps represent each sender's travels.

We did not have to implement the ML model for this experiment as the code was available. Instead, we added SA to the process to see the effect on the ML model's final results. Figures 3.4, 3.5, 3.6, and 3.7 show the obtained results (for the KNN, SVM, KNN with SA, and SVM with SA) after experimenting on a local machine. We did not find a significant difference between the published results and the obtained results. However, we had to compare with the newly obtained results to measure the impact of SA on them. So et al. [56] represented two experiments, one using the

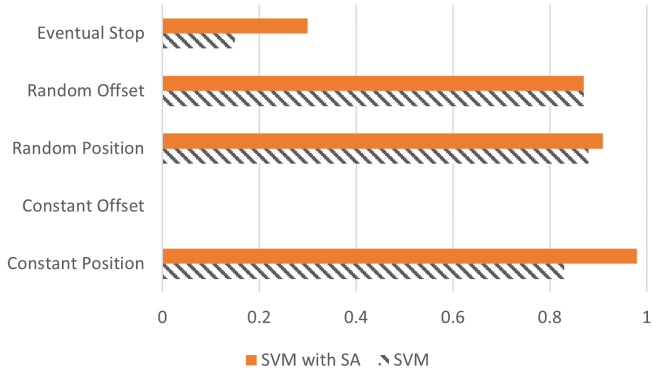


Figure 3.6: Recall comparison between the obtained results by the SVM model and after integrating SA

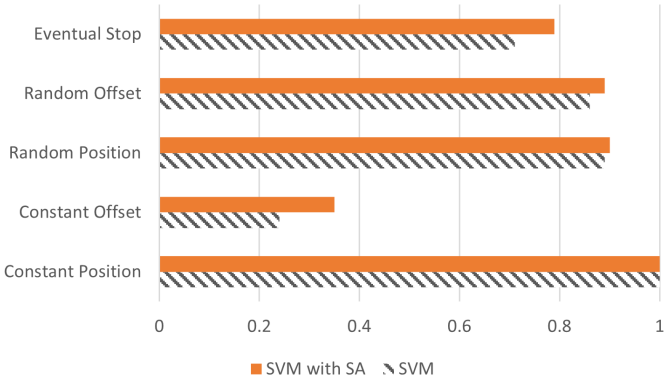


Figure 3.7: Precision comparison between the obtained results from the SVM model and after integrating SA

KNN model and the other using SVM. For the KNN model, Figures 3.4 and 3.5 show the increase in recall, especially for Attacks 4 and 5, where they increased the recall by 36% and 52% with only 3% and 11% decrease in precision. Furthermore, for the SVM model, Figures 3.6 and 3.7 show how we increased recall and precision in all attacks.

Figures 3.4 and 3.6, show a noticeable small recall rate in the constant offset attack both for the KNN and SVM models. Nevertheless, this is a detection problem

in the first place. Looking at how the same plausibility checks have been used for both models, we can see how we ended up with the same behavior regarding detecting the same attack (constant offset attack). Accordingly, better plausibility checks for detecting this attack are needed in general. Still, under those circumstances, we can see that SA maintained the same behavior of improving the precision rate when the recall is saturated, as shown in Figures 3.5 and 3.7. Ultimately, SA can only improve recall when there is room for improvement in the first place. If the ML model cannot detect the attack at an acceptable rate from the beginning, this limits SA capabilities in enhancing the final results.

3.6 Summary

Misbehavior attacks are one of the biggest challenges to achieving secure and reliable VANETs. Additionally, ML models, with all their advantages and preeminence, require a new perspective to perform better and acclimate with VANETs characteristics. One of those characteristics is the ability to share personal information using BSMs in an open environment. In this work, we used a publicly available dataset, VeReMi. We showed how utilizing SA with traditional ML model approaches can lead to better detection of misbehavior attacks. Furthermore, LT was used to measure SA. By incorporating SA with ML, we ascertained that the ML model's classification results still have room for improvement (by accepting a message labeled as an attack and vice versa) to obtain a better detection rate. Moreover, we found that SA increases the network sensitivity to detect misbehaving. Accordingly, we noticed how this manifested in a better recall rate, which eventually increased the detection rate.

The performance of the introduced system was evaluated through extensive experiments. The proposed system showed promising results with over 50% increase in recall in some cases. It has been evident that utilizing SA with ML for misbehavior detection in VANETs is promising for achieving a better detection rate and keeping the network secure and reliable.

Chapter 4

Dataset for Attack Endgame Prediction

Within the context of the proposed TMS, proactive approaches such as Situational Awareness (SA) play a pivotal role, primarily by facilitating the prediction of trustworthiness concerning trust subjects. However, we found out that pursuing our objectives for the remaining components of the TMS encounters significant obstacles due to the constraints imposed by the currently available datasets for VANET security.

In the preceding chapter, the exploration into VANET security revealed significant limitations in the existing datasets. These limitations include their inability to provide enough data for attack prediction and to evaluate proactive security techniques. This chapter introduces a novel dataset named VeReMi for Attack Prediction (VeReMiAP) [18]. The introduced dataset is specifically designed for proactive security in VANETs, focusing on predicting the impact of attacks.

4.1 Introduction

The development of this dataset is not merely a response to the identified gaps but a strategic move towards enhancing the field of VANET security. By creating a dataset

that is inherently proactive-friendly, we aim to facilitate the development and rigorous evaluation of proactive security solutions within VANET environments. This chapter outlines the methodology behind the generation of this dataset, emphasizing its potential to revolutionize how security measures are conceptualized, implemented, and assessed in VANETs. This new dataset is a critical tool for researchers and practitioners, offering a more nuanced and practical approach to preemptively addressing security challenges in VANETs.

Transitioning into the specific functionalities and characteristics of VANETs, it is pertinent to understand the types of communications integral to their operations.

As mentioned in 2.1.1, vehicular networks predominantly utilize DSRC, which facilitates the exchange of two key types of messages: BSMs and CAMs. BSMs are real-time, succinct vehicle transmissions that convey essential data such as location, velocity, and heading. CAMs provide real-time information about the vehicle and encompass additional data such as road conditions, traffic congestion, and incidents. In this context, given VANETs' dynamic nature and the critical need for road safety, securing these messages emerges as a major concern.

The susceptibility of VANETs to misbehaving and insider attacks poses a unique challenge, distinct from other network types. Therefore, ensuring the security of VANETs and the integrity of BSMs and CAMs is essential for maintaining the reliability and trustworthiness of the transmitted information. Implementing security prediction methods and evaluating them with datasets designed for this purpose is integral to enhancing the overall security and reliability of the network and its associated applications.

In cybersecurity, attack prediction is a method used to warn of a possible attack,

allowing for adequate time to take precautions or measures to minimize the attack's impact. Several different techniques can be used for attack prediction [68, 69], including:

- Anomaly detection uses algorithms to identify unusual patterns in data that may indicate an attack.
- Behavior modeling involves analyzing the user or vehicle's behavior patterns to identify potentially malicious activity.
- Risk scoring assigns a score to each attack based on the likelihood and potential impact.

We focus on simulating and predicting the effect of the attack. Thus, we adopt the behavior modeling approach from the above list of techniques. Behavior modeling can play a crucial role in generating a dataset designed to simulate the effect of an attack in VANETs. This process involves creating a representation of the typical behavior patterns of the network's participants, vehicles, and RSUs. The analysis and modeling of vehicle behavior in VANETs are essential in detecting deviations that may indicate the presence of an attack. This information can be leveraged to create datasets that simulate the impact of attacks on the network, and those datasets can be used to assess various security measures and evaluate their effectiveness.

To demonstrate the severe impact that an attack can have on road conditions and the surrounding vehicles, we identify the attack class of reporting fake information as a major threat in VANETs. We simulate and include the FRA in the dataset. By doing so, we aim to showcase the real danger posed by this type of attack and highlight the need for effective countermeasures to mitigate the risk of such attacks in VANETs.

Including FRA is a crucial step toward understanding the complex nature of VANET security and developing robust security solutions that can effectively protect against this threat.

The work in this chapter makes several contributions to the field of VANET security. First, we introduce a new dataset focusing on attack prediction rather than just detection. Second, we extend the Framework For Misbehavior Detection (F2MD) [70] by incorporating CAMs, enabling the exploration of new types of attacks. Third, we introduce a new class of attacks related to CAMs, the FRA. We simulate a specific scenario of this class of attack where a vehicle reports a fake accident and incorporate the effect of the attack, which was an occurrence of a hazard, into the dataset. Finally, we conduct temporal, geospatial, and behavioral analyses of the dataset to understand the data better.

The rest of the chapter is organized as follows. Section 4.2 discusses related work on existing datasets and attack prediction in VANETs. The work to extend F2MD for enabling the generation of the new dataset is presented in Section 4.3. Section 4.4 describes the setup of the simulation. Section 4.5 provides a detailed description of the dataset generation process. Section 4.6 highlights the simulation results and provides an in-depth analysis and evaluation. Finally, Section 4.7 provides a summary of the chapter's work.

4.2 Related Work

Creating datasets for simulating cyber threats in VANETs has long been challenging for researchers. However, some datasets cover inter-vehicle and intra-vehicle attacks.

The approaches used by researchers to build these datasets typically involve a combination of real traces and simulations. The VeReMi dataset, introduced by Van der Heijden et al. [16], was created to address the lack of a standard dataset for MDSs and to establish a baseline for evaluating MDS performance. The F2MD framework generated VeReMi and VeReMi Extension [71]. The dataset consists of vehicle message logs and primarily BSMs. VeReMi includes various types of attacks in VANETs, but some attacks are not covered.

Swessi et al. [72] generated a synthetic dataset for VANET intrusion detection in the inter-vehicle network environment. The dataset was created using the network simulator NS-3 and the unique characteristics of VANETs, such as high host movement speed and rapidly changing topology. The Swessi et al. dataset simulated ten scenarios of VANET host mobility and included five types of security threats. The user had the flexibility to choose the number of vehicles and the number of malicious nodes in the network; this allowed for a high degree of realism and specificity in the training of VANET intrusion detection systems. Swessi et al. generated a synthetic dataset that provides a valuable tool for researchers and developers to better understand the host's behavior in the inter-vehicle network and design effective intrusion detection systems.

The Next Generation Simulation (NGSIM) dataset [73] offers valuable insights into real-world traffic patterns by capturing detailed vehicle trajectory data using synchronized digital video cameras. The U.S. Federal Highway Administration transcribed the data using the custom-developed NGVIDEO software application. Coifman and Li introduced dynamic noises into the data to simulate misbehavior [74]. Researchers [75] utilized the NGSIM dataset to develop an effective misbehavior detection model

based on ANN techniques. However, despite providing precise vehicle location data, the NGSIM dataset is limited in terms of its time-space coverage, the variety of data collected, and the lack of specific attacks being represented.

Some researchers introduced proactive solutions for attack prediction in VANETs [76, 12]. These efforts demonstrate the potential for developing effective solutions to address the lack of datasets in the field. For example, in their work, which we mentioned before in Chapter 2, Dasgupta et al. [12] employed an LSTM model to detect spoofing attacks in a real-world driving dataset, comma2k19 [39]. The dataset includes sensor data such as the Control Area Network (CAN) [77], GNSS [78], and IMU [79], which were used to predict the distance between consecutive locations. The authors focus on a specific type of spoofing attack where GNSS signals are manipulated to mislead the autonomous vehicle. The LSTM model shows a high detection rate with low error and latency, which is suitable for VANETs' constraints. However, the full impact of these efforts remains unknown without a complete dataset to evaluate prediction capability.

4.3 Preliminaries

The F2MD has undergone substantial enhancements to address evolving challenges in VANET security. This section highlights the modifications, including integrating behavioral analysis techniques and CAMs to enable more accurate attack prediction. Table 4.1 provides a comparison between the traditional F2MD and the extended F2MD. The traditional F2MD served as a framework for VANET security research, offering basic capabilities for security evaluation. However, traditional F2MD lacked

	Traditional F2MD	Extended F2MD
Framework Overview	Provides a framework for VANET security research.	Extends the framework to address evolving security challenges.
Behavioral Analysis	Does not incorporate behavioral analysis techniques.	Integrates behavioral analysis to enhance attack prediction.
CAMs	Does not include CAMs.	Incorporates CAMs for a realistic and comprehensive dataset.
Dataset Enhancement	Limited dataset capabilities for attack prediction.	Enhanced dataset with attack impact to evaluate prediction techniques.
Attack Effect Prediction	Less effective in predicting attack's effect in VANET.	Enables attack effect prediction in VANET.

Table 4.1: Comparison between traditional F2MD and extended F2MD

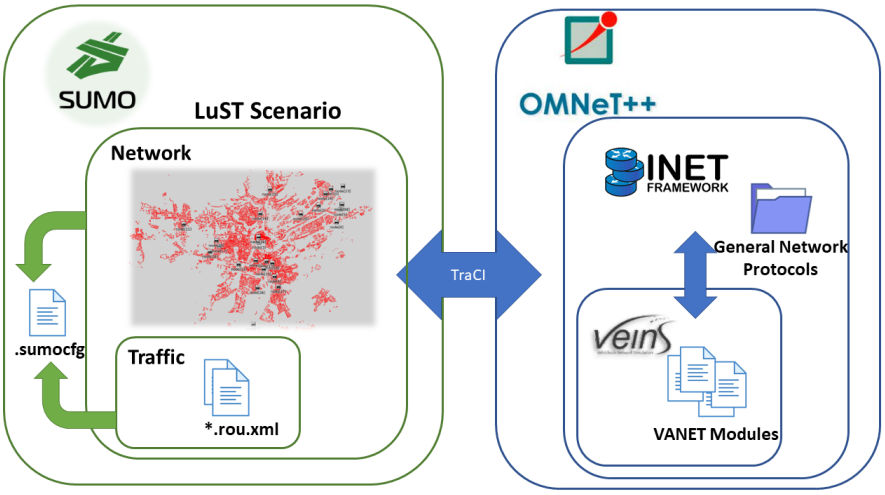


Figure 4.1: Interconnectivity of frameworks in F2MD for dataset generation

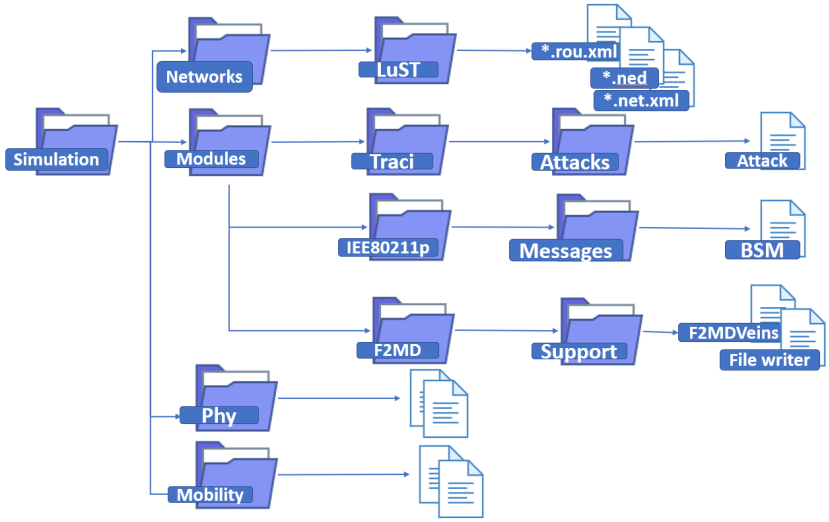


Figure 4.2: File structure of the extended F2MD framework: displaying the modified and added files for new extension

the incorporation of behavioral analysis techniques and CAMs, which limited its effectiveness in predicting sophisticated attacks and generating comprehensive datasets. In contrast, the extended F2MD addresses these limitations by integrating behavioral analysis to enhance attack prediction and incorporating CAMs to provide a more realistic dataset. Extended F2MD empowers researchers with advanced analysis capabilities, enabling the evaluation of complex attacks and the development of robust attack prediction approaches.

4.3.1 An Overview of the Framework for Misbehavior Detection

The F2MD [70] is a framework designed to support research in VANET security. One of its capabilities is generating datasets, including VeReMi [16] and VeReMi Extension [71] datasets have been applied in various security studies [67, 14]. The developers of F2MD made the source code publicly available to encourage its use by

other researchers. F2MD comprises various modules and components, each responsible for functionalities such as vehicle and mobility, attack generation, and network implementation.

We leveraged F2MD in two ways. First, we extended its capabilities by incorporating a new class of attacks and their associated effects into the generated dataset. Second, we utilized F2MD to generate a dataset that serves as a resource for evaluating security prediction techniques.

Figure 4.1 illustrates the interaction of the various components incorporated within F2MD to deploy the simulation and generate VeReMiAP. The figure shows two key elements of the simulation process. The first element, the Simulation of Urban MObility (SUMO) [80] tool, is utilized to create realistic network mobility traffic. The generated traffic is then imported into OMNeT++ in the second element, where the network traffic is generated utilizing both VEINS and the INET [81] framework. The Traffic Control Interface (TraCI) [82] is utilized for real-time access and manipulation of road traffic simulations. Finally, the collected data undergoes processing and is exported as VeReMiAP.

Figure 4.2 displays a segment of the F2MD file structure, specifically highlighting the files that underwent modifications (indicated by their names in the figure). Given the extensive size of the F2MD file structure, it was not feasible to include every component in the figure.

4.3.2 Leveraging Behavioral Analysis for Enhanced Attack Prediction

To enhance the compatibility of VeReMiAP with attack prediction, we utilized behavior modeling to identify the necessary variables to be included. In VANET, by

analyzing the behaviors of different types of vehicles and their interactions, we identified patterns that can be used to anticipate attacks. Our objective in the process of generating VeReMiAP is to fulfill three behavior modeling techniques:

- Analyzing log data: Logs from sources like networks, applications, and user activity can reveal patterns that may indicate a security breach.
- Monitoring user activity: Real-time monitoring of user behavior can indicate signs of potential system compromise.
- Analyzing user behavior: By analyzing user behavior over an extended period, it is possible to identify changes in behavior that may indicate an attempt to compromise the system.

The extension of F2MD provided the necessary data to support these techniques, aiding the prediction process. First, to enhance the accuracy of behavior pattern detection, we collected extensive data from each vehicle to create more detailed logs. Second, we integrated BSMs and CAMs to complete an overview of vehicle activities. Last, we included the impact of the attacks, occurring at different times than the actual attack, to assist in analyzing vehicle behavior patterns over time and to facilitate the prediction process.

4.3.3 Integrating CAMs into the extended F2MD Framework

We have implemented CAMs into VeReMiAP. In Figure 4.2, we observe a modified BSM file and other related files in the framework that have been adjusted accordingly. Both types of messages can be broadcast between vehicles in VANET. Including CAMs in VeReMiAP allows for training and evaluating machine learning models to

predict attacks in the network, thereby improving the effectiveness of attack prediction and prevention strategies.

4.4 Scenario Setup

This section explains how the simulation was set up and how the dataset was generated.

4.4.1 Traffic Model

In this subsection, we delve into the foundational elements of the traffic model, which is constructed using two primary components: the traffic simulation software, SUMO, and the traffic scenario, LuST. The choice of these tools is grounded in their widespread adoption in academic and practical applications, attributed to their versatility and the realistic behavior of generated vehicles.

Network and Traffic

We used SUMO to simulate vehicle mobility. The SUMO simulation environment involves defining the network and traffic aspects. The network is specified in a .net.xml file, which outlines the layout of roads, intersections, and other relevant features within the simulated scenario. The traffic in SUMO is defined in one or more .rou.xml files that specify the vehicle routes, departure times, and other important parameters. As shown in Figure 4.1, in a SUMO simulation, connecting the network and the traffic requires the specification of both the .net.xml file, and the .rou.xml file(s). This connection is made through a .sumocfg configuration file, which instructs SUMO to use the designated .net.xml file with the "–net-file" option and .rou.xml file(s) with

the "–route-files" option.

LuST Scenario

The Luxembourg SUMO Traffic (LuST) Scenario [83] was chosen for this research because it is a highly realistic VANET simulation. The LuST scenario includes real-world data and features a diverse range of vehicles and roads, such as intersections, roundabouts, and highways, creating a comprehensive and realistic scenario for VANET research.

4.4.2 Communications

To ensure standardization and compatibility in the communication between vehicles, particularly for the implemented CAMs, we followed the ETSI EN 302 637-2 standards for CAMs. ETSI specified that CAMs should be generated based on the vehicle's kinematics within the time boundaries of $T_{\min}$ and $T_{\max}$ [84].

$T_{\min} = 100$ ms, which sets an upper limit on the maximum rate of CAM generation at 10 Hz, meaning that the time interval between any two CAMs cannot be shorter than T_{min}.

$T_{\max} = 1000$ ms, which sets an upper limit on the minimum rate of CAM generation at 1 Hz. In order to comply with the IEEE 802.11p MAC specification [85], transmitting the generated CAMs on a dedicated channel was necessary. The implementation of F2MD followed the IEEE 802.11p MAC standards, which provide several PHY layers and a common MAC sub-layer for vehicle-to-vehicle communication.

Parameter	Value
Mobility	SUMO
Mobility scenario	City streets
Simulation area	2300m x 5400m
Simulation duration	3600s
Threat	Fake reporting attack
Attack probability	0.5
Number of vehicles	826
Exchanged messages	BSMs, CAMs
MAC implementation	802.11p
Beaconing rate	1 Hz
Speed	Maximum speed limit

Table 4.2: Simulation parameters

4.4.3 Attack Model

Our scenario included malicious and benign vehicles to emulate potential cyberattacks. The malicious vehicles execute malicious actions by transmitting false messages about themselves or the road conditions. The benign vehicles followed predetermined routes and communicated with each other, replicating the behavior of vehicles in a typical VANET environment.

We conducted a simulation that included FRA and specifically targeted CAMs. This attack differs from the previously introduced attacks in the F2MD framework that were focused on BSMs. This endeavor was challenging, as it required meticulous utilization of the available functionalities provided by INET and SUMO. In the simulation, INET and SUMO frameworks collectively served as valuable data

sources, providing the necessary information typically collected from onboard sensors or exchanged by RSUs. This information encompassed the vehicle's moving direction (an example of an onboard sensor type information) and the lane index and current road maximum deceleration (examples of RSU-received type of information). This collaboration between INET and SUMO played an important role in providing accurate and representative data, improving the authenticity and realism of the simulated scenarios.

CAMs report road conditions, providing valuable information to vehicles within the network. CAMs expand upon the traditional BSMs fields by incorporating additional fields that improve the comprehensiveness of the information exchanged. These supplementary fields include the event ID, road ID, lane position, lane index, X and Y-axis directions, maximum deceleration, and maximum speed. The event ID in CAMs denotes the specific type of event being reported, such as accidents, road construction, or roadkill. Including fields such as, road ID, lane position, and lane index allows for a precise definition of the location of the reported event. Since vehicles on the same road may travel in different directions, the CAMs also encompass directional components along the X and Y axes. Furthermore, critical information such as maximum deceleration and maximum speed is integrated into CAMs to detect abnormal speeding or deceleration behaviors. Great vehiclee was taken to select and include the most relevant and correlated information that accurately represents various events. Through this simulation and incorporating pertinent information into the dataset, we could emphasize the significance of CAM-related attacks and their potential impact on VANETs.

We maintain the same attacks from the F2MD attack library as used in the

VeReMi dataset [16] and further supplemented the library attacks with a specifically designed attack and its effect. We introduced a CAMs-related attack called FRA. Our scenario involved malicious vehicles falsely reporting an accident, which could potentially cause a real accident or hazard.

Attack Trigger: The FRA is initiated based on the attacker's discretion, as they choose when to trigger it based on their convenience. In the simulations, the attack is randomly initiated.

Attack Manifestation: The FRA manifests when a malicious vehicle intentionally transmits misleading information about traffic or road conditions, prompting other vehicles to respond defensively. The FRA, which was included in the simulations, exhibited its potential repercussions on road conditions and the safety of surrounding vehicles, underscoring its prominence as a significant threat in VANETs.

Attack Frequency: The FRA attack frequency adheres to the standardized frequency guidelines outlined in ETSI EN 302 637-2 for CAM transmission. For the simulation, we enforced the upper limit on the maximum rate, which is 10 Hz for CAM generation.

Attack Effect: In order to implement the effect of the simulated FRA attack, we added the code to simulate the actions taken by the vehicles in response to receiving BSMs or CAMs. We used the mobility module in F2MD to simulate a malicious or benign vehicle slowing down and gaining speed. The *slowDown*() function provided by the mobility module allowed the vehicle to lower its speed to a threshold and gradually increase, returning to the maximum, simulating real-life behavior. The effect of an FRA attack on the road is a traffic hazard. To record the occurrence of a hazard, we defined specific conditions in the simulation. We defined the occurrence

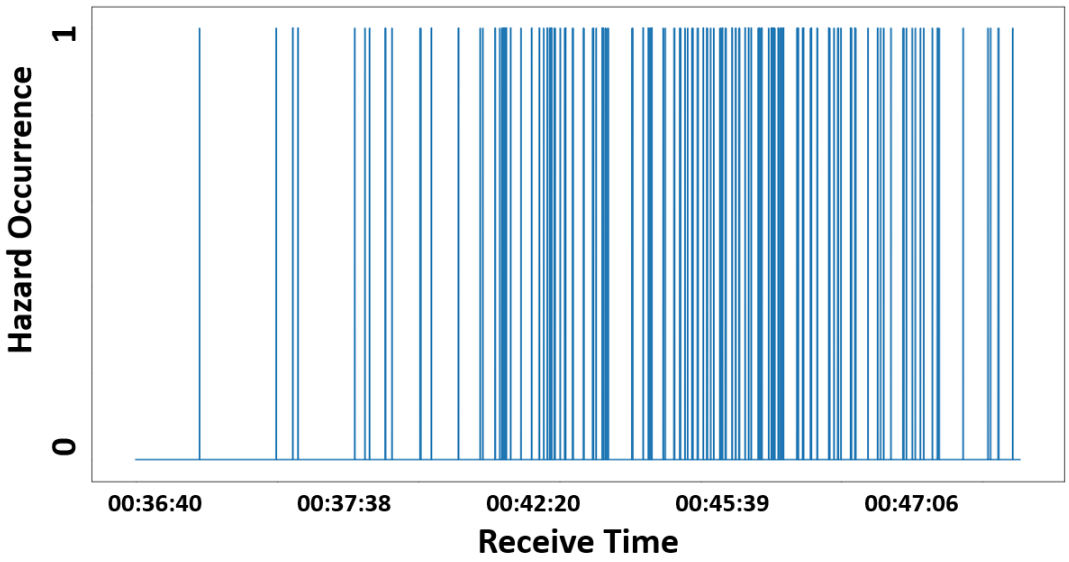
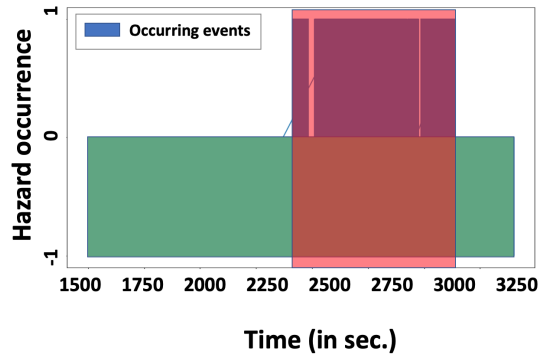


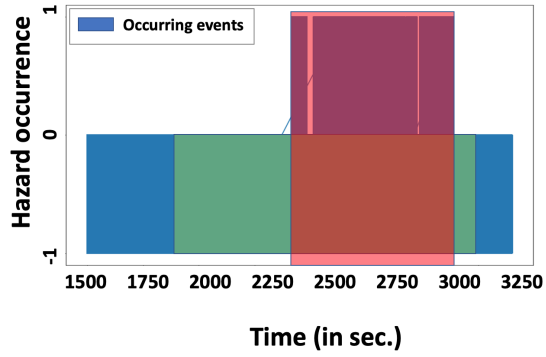
Figure 4.3: Occurrence of hazards over time, where each vertical line indicates the specific time at which a hazard occurred

of a hazard being when a vehicle is driving on the reported road, the vehicle's speed drops below 20%, and the number of vehicles on the road exceeds five. These numbers are based on experimentation and were used to simulate realistic conditions.

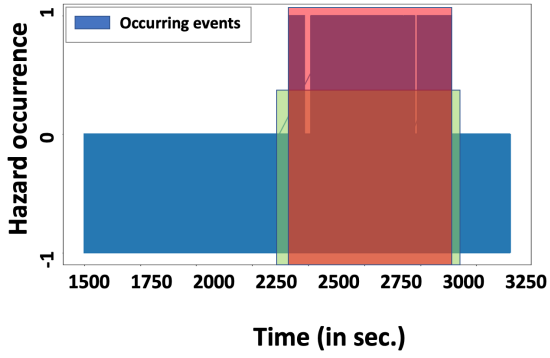
Figure 4.3 shows a zoomed-in figure for the occurrence of a hazard over a period of time. The figure shows spikes in the frequency of hazards at specific points in time, with each spike representing a hazard occurrence. Additionally, the figure shows how we can create a hazard and then allow the simulation to get back to its normal state multiple times without crashing. Our simulation also provides an indicator of the hazard occurrence time. It is important to note that the time at which the attack message was sent will be ahead of the actual occurrence of the hazard. This is because it takes time for the hazard to materialize once the malicious message has been received. Figure 4.4 provides an overview of the events that transpired during the simulation period. The binary notation used in the three graphs assigns a value



((a))



((b))



((c))

Figure 4.4: Assessment of the impact of data scope on hazard prediction, with each subfigure displaying a different scenario in which a different amount of data is included

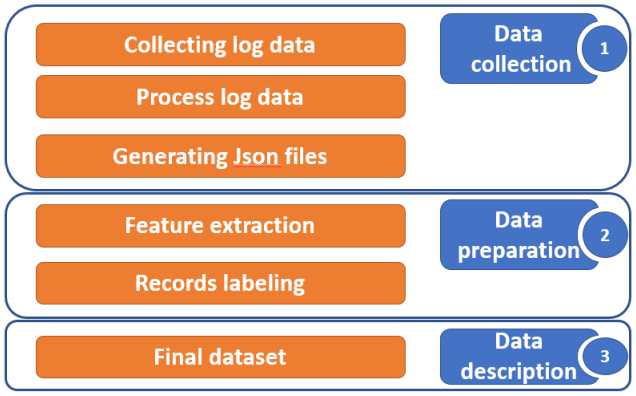


Figure 4.5: Steps for generating the dataset

of 1, representing the occurrence of a hazard at a specific time, a value of 0, indicating that no hazard was detected at that time, and a value of -1, indicating the presence of an unidentified event. A thorough analysis and discussion of the contents of Figure 4.4 will be presented in Section 4.6.4

The simulation in the system model was run for one hour, during which the network traffic and the state of each node were recorded. The resulting dataset contains detailed information about each node's location, communication, and attack status at different time intervals. This approach enabled us to create a semi-realistic dataset that captures VANET behavior under the FRA.

4.5 Dataset Generation

In this section, we outline the steps involved in generating the dataset. Figure 4.5 illustrates the entire dataset generation process to ensure the quality and consistency of the data collected.

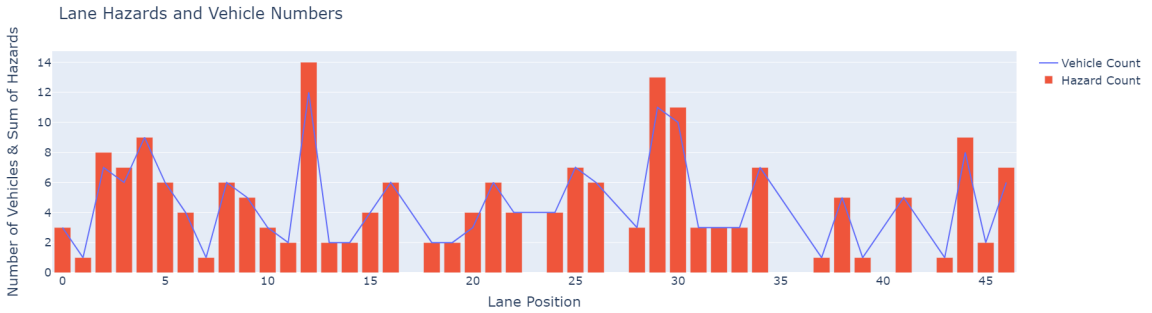


Figure 4.6: Shows the correlation between the occurrence of hazards and the number of vehicles on the road.

4.5.1 Raw Data Collection

The generated synthetic dataset was created using a combination of existing frameworks and simulation techniques. The primary data source was F2MD, which provided the basic road network and vehicle information. This information was then used as a starting point for simulating an attack within VANET.

We leveraged the capabilities of F2MD to include the implementation of CAMs, incorporating a novel FRA, and capturing the impact of the attack on the VANET. The simulation scenario was defined by a set of input parameters outlined in the scenario .ned file. These parameters, including configuration settings and model specifications, listed in Table 4.2. The primary inputs included the simulation start and end time and the type and the intensity of the attacks. The intensity of the attacks was defined as the percentage of malicious vehicles in the simulation, allowing for fine-tuning the attack level.

After running the simulation, we collected the generated BSMs and CAMs transmitted between vehicles. To efficiently organize and access the data, we converted

the VeReMi writing module into JSON files, creating a separate file for each vehicle that included information on the communication with surrounding vehicles. Our custom code, featuring distinct message codes for BSMs and CAMs, ensured that the additional information in CAMs was recorded for future analysis. Using JSON files ensured that the data was stored in a widely recognized and accessible format, making it easier for future research and analysis.

4.5.2 Dataset Preparation

Data preparation is an essential step in the dataset implementation process as it ensures that the data is in a suitable format and meets the requirements for the intended use, leading to accurate and effective results. Therefore, we converted the data files from JSON to CSV format to facilitate the subsequent data preparation steps.

Feature Extraction

The raw data collected comprised various parameters such as vehicle speed, acceleration, and location. To facilitate the analysis and prediction process, we extracted features from the raw data, including road number, maximum speed limit, and type of attack. These features were chosen to provide a representation of the road network and the impact of cyberattacks on the VANET.

To ensure the authenticity and realism of the simulated scenarios, we included information that closely resembles real-life situations such as road speed limit, lane position, and current direction. This information was accurately incorporated into the CAMs to enhance the realism of the dataset. Additionally, RSUs can contribute

important details such as lane index and the maximum deceleration limit of the current road, which were also taken into account during the simulation. The RSU and onboard sensor-related information in the simulation was obtained from the SUMO libraries.

Record Labeling

The labeling process involved defining the attributes of each feature, such as data type and range of values, which ensured that the data was consistent and well structured. The most important features we added were the new FRA and its effect to comprehensively represent the simulated road network and the impact of cyberattacks. The messages were labeled with a 0 for benign and 1 for attack, while the occurrence of hazards at each timestamp was labeled as 0 for no hazard, 1 for the occurrence of a hazard, and -1 for undefined to account for the uncertainty of the network state.

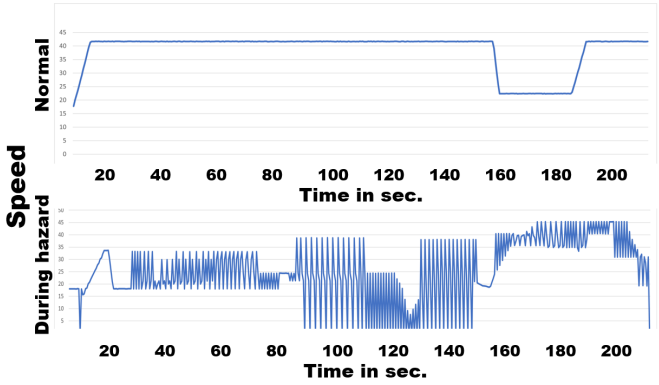


Figure 4.7: Comparison of vehicle speed with and without hazards over time

4.5.3 Dataset Description

The synthetic dataset used in this work is publicly available on Borealis [18] under the name VeReMi for Attack Prediction (VeReMiAP). The dataset comprises 2 gigabytes

Field	Description
type	Message type (BSMs or CAMs)
rcvTime	Message reception time
vehicleId	Vehicle identifier (receiver)
pos	Vehicle's position (X, Y, Z coordinates)
pos noise	Position noise (X, Y, Z coordinates)
spd	Vehicle's speed (X, Y, Z coordinates)
spd noise	Speed noise (X, Y, Z coordinates)
acl	Vehicle's acceleration (X, Y, Z coordinates)
acl noise	Acceleration noise (X, Y, Z coordinates)
sender GPS	GPS coordinates of the message sender
exchangedMessageType	Type of the exchanged message
EventID	CAMs event identifier
EventType	CAMs event type (e.g., accident)
RoadID	Road identifier
hazardOccurrence	Indicator of hazard occurrence [0,1]
hazardAttack	Indicator of hazard attack [0.1]
laneIndex	Lane index if the road has multiple lanes
lanePosition	Starting position of the detector on its lane, measured in meters from the lane's start
maxSpeed	Maximum allowed speed on the current road
currentDirection	Vehicle's direction (X, Y, Z coordinates)
maxDeceleration	Maximum deceleration
sender	Message sender
hazardOccurrencePercentage	Percentage likelihood of hazard occurrence

Table 4.3: Comprehensive overview of dataset field definitions

of data, represented as JSON files. These files capture the exchange of messages between vehicles in the network. The messages were organized chronologically, and each file contained BSMs and CAMs. The interval between each message was set to one second. Each record corresponds to a BSM or a CAM in the dataset and encompasses the standard data fields in each message.

Table 4.3 provides a description of the dataset fields. It is important to note that these fields are presented in the JSON files, while the CSV files contain a subset of the

fields. Therefore, researchers can select the appropriate fields based on the specific requirements of their analysis, ensuring optimal performance and relevance to their research objectives.

In addition to the standard BSMs' data fields, the dataset includes two fields specific to the work's objective. The first field, Fake Accident Attack, indicates an attack message that transmits false information. The second field, the Attack Effect, indicates when the corresponding hazard event occurs. It is worth noting that the attack effect field is not directly associated with the attack message in terms of timing, as the messages themselves serve as timestamps to identify the hazard occurrence.

To accurately reflect real-world conditions, the dataset presented an unbalanced distribution of attack and benign messages. The fake accident attack was implemented with distinct features, one was the low number of attack messages compared to the overall number of exchanged messages and the low time of the hazard occurrence compared to the total overall time where the network state is normal. As mentioned in [86], the assumption that the majority of vehicles are truthful is based on the difficulty of executing collision attacks on a large scale in practice. This disparity creates a challenge, as the imbalance may negatively impact the performance of the prediction models.

To address this disparity challenge, we used two approaches. The first approach was to increase the number of fake reporting messages, but this would not accurately reflect the real-life scenario and would normalize the occurrence of fake messages throughout the network. The second approach was to leave it to different data imbalance handling techniques such as oversampling, undersampling, or costsensitive learning. The decision was made to choose the second option; even though it results

in an imbalance as well, it maintains the real-life scenario and will provide a better evaluation of the prediction models. This factor is important, as any solution that does not consider the imbalance in the dataset will not perform as expected in a real-world scenario.

In summary, the VeReMiAP dataset contributes to the research community by providing a realistic resource for investigating various aspects of VANET security. Researchers can leverage this dataset to address questions related to attack detection, prediction, and mitigation in VANETs. VeReMiAP enables the analysis of an attack's impact on network performance, the evaluation of existing security mechanisms, and the development of novel approaches to enhance VANET security. Additionally, VeReMiAP expands upon existing resources by incorporating diverse attack types, including fake reporting attacks. By providing detailed information about the attack parameters, network conditions, and vehicle behaviors, researchers will be able to examine the complexities of VANET security and contribute to advancing effective security measures in this domain.

4.6 Analysis and Evaluation

4.6.1 Data Analysis

This section presents the results of the analysis of VeReMiAP. We applied temporal, geospatial, and behavior analysis techniques to identify potential attack patterns and provide insights for developing effective strategies to mitigate the impact of attacks on VANETs. The findings and patterns observed in the dataset can serve as a valuable resource for researchers in developing robust security measures and countermeasures to safeguard VANETs against cyber threats.

Temporal Analysis: Hazard Impact on Vehicle Speed Over Time

The temporal analysis was conducted to observe the variation in vehicle speed over time, particularly when it is in hazardous conditions compared to normal traveling conditions. The results, depicted in Figure 4.7, reveal a meandering pattern in vehicle speed, characterized by repeated dips to zero and broad fluctuations in speed. This meandering pattern indicates that the presence of a hazard impedes the ability of the vehicle to maintain a consistent speed, leading to an unsteady speed trend over an extended period. The temporal analysis helps us understand the impact of hazards on vehicle speed, making it a valuable tool for evaluating time-series data.

Geospatial Analysis: Fake Messages and their Effect on Road Conditions

Geospatial analysis is a method of analyzing data that uses geography and spatial information to identify patterns, trends, and relationships within the data. In VANETs dataset that contains information about hazards and the number of vehicles on the road, geospatial analysis can identify the distribution of hazards located on the road and the relationship between the number of vehicles and the occurrence of hazards. This distribution can be achieved by mapping the location of hazards and the number of vehicles on a geographic map and analyzing the spatial patterns and relationships between the two variables. Geospatial analysis can also identify the areas of hazard occurrences, such as areas with high traffic density or specific road conditions. We can develop safety measures to prevent hazardous incidents by analyzing data in a spatial context.

Figure 4.6 shows the correlation between road lanes and the concentration of vehicles during a hazard event. The bars in the figure represent the accumulation of

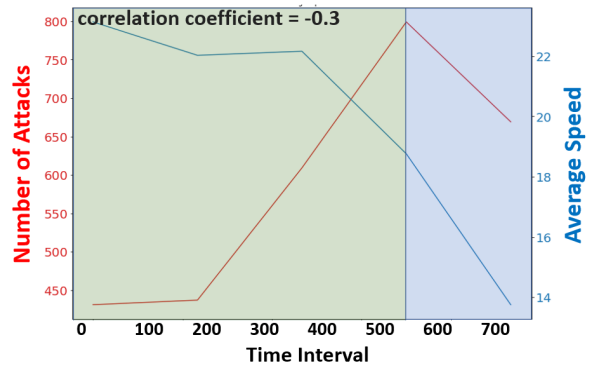
hazard timestamps in the dataset, and the blue line indicates the number of vehicles. The hazard accumulation and the number of vehicles were recorded based on the lane where the hazard occurred; this is identified where hazards are most likely to occur on the road and how many vehicles could be impacted. This information can be used to anticipate and prepare for future hazards caused by a FRA.

Behavior Analysis: Effects of Attack on Network Performance

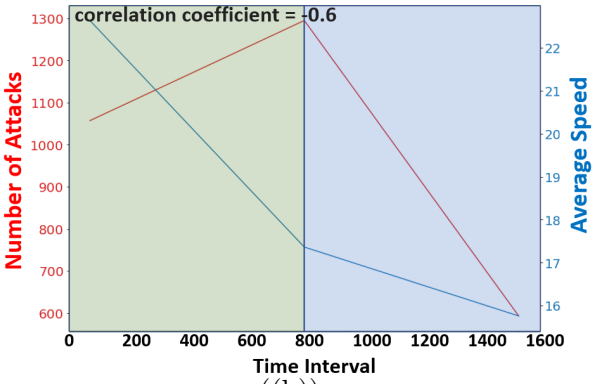
Behavior analysis in VANET datasets refers to the examination of the behavior of vehicles in the network over time. In VANETs, this analysis can be used to evaluate the effect of attack messages on the average speed of vehicles. By monitoring changes such as vehicle speed, acceleration, and location, we can identify patterns and trends in the behavior of vehicles in response to attack messages.

The graph in Figure 4.8 provides insights into the effect of received attack messages on the average speed of various types of vehicles in the network. The figure displays the number of attack messages and the overall vehicle speed at a given time. Three subgraphs, each representing a different experiment, are included in the figure to provide a complete understanding of the behavior of the network at different time intervals. The experiments were conducted by gradually increasing the time interval to obtain a more generalized view of the network behavior and observe the patterns in the dataset.

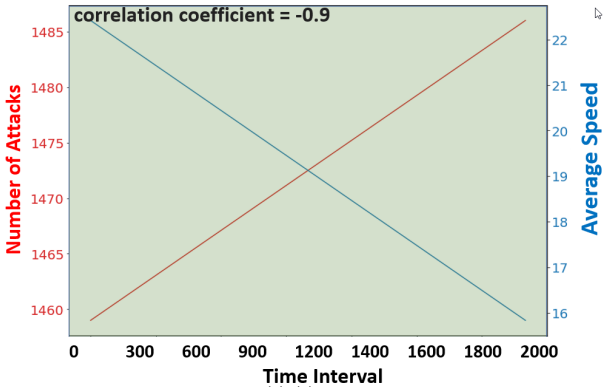
In Figure 4.8(a), based on a 700 s time interval, a reverse correlation between the number of attack messages in the network and the average vehicle speed was observed, with a correlation coefficient of -0.3. However, the blue highlighted area indicates anomalous behavior after the peak in the number of messages, which may



((a))



((b))



((c))

Figure 4.8: Correlation between attack messages and average network speed over time, with highlighted regions of correlation

result in a misinterpreted positive correlation. A more extensive view was necessary to gain a more thorough understanding of the network's behavior. In Figure 4.8(b), with a 1500 s time interval, the correlation between the number of attack messages and the average speed becomes more apparent with a correlation coefficient of -0.6.

Figure 4.8(b) indicates that increasing attack messages decreases average speed. As the number of attack messages peaks, the network experiences road hazards. The blue highlight in the figure demonstrates that even though the number of attack messages decreases, the average speed is not recovered, resulting from the road hazard rather than the impact of the attack messages. While Figure 4.8(b) provides a better understanding of overall behavior, a more extensive view is still necessary.

The impact of attack messages on the average speed of vehicles in a VANET network is depicted in Figure 4.8(c), which was based on a 2000 s time interval. The figure provides a view of the relationship between the number of attack messages and the average speed. It shows a clear reverse correlation, as indicated by a correlation coefficient of -0.9. This reverse correlation highlights the overall behavior of the network, including the smaller-scale behaviors observed in previous figures. The previously observed blue-highlighted area in Figure 4.8(b), where the vehicle speed does not recover even though the attack messages decrease, is not present in Figure 4.8(c), showing that the overall network behavior between the vehicle speed and attack messages is a reverse correlation. By analyzing the network behavior across different periods, we gain a deeper understanding of the network's behavior, both in the short and long term. This information is valuable for evaluating security measures and planning future research and development efforts in VANETs, as it highlights the significant impact that attack messages can have on vehicle behavior.

4.6.2 Predicting the Effect of the Attack

In the following experiments, we aim to predict the effect of a false traffic warning message on VANETs. At a designated time, malicious vehicles will transmit a falsified traffic alert. However, the full ramifications of this false information may take some time to become apparent. What we are looking at is the aftermath of a malicious vehicle's false traffic warning being received.

In our experiments, we use a Recurrent Neural Network (RNN). RNN is incorporated with memory to store information from prior inputs and to influence both input and output. Since input and output are not independent of each other, memory-based models such as RNN are more suitable.

We conduct three experiments to evaluate attack endgame prediction modules. We consider the results obtained by the LSTM and GRU models as a baseline. The first two experiments create a baseline for NN models. The first experiment is to determine how far we can predict the future. The second experiment evaluates the correlation between the type of carried-on information from past interactions and prediction results. The third experiment evaluates the proposed model against the baseline.

Using the Train-Valid-Test split technique, we implement a cross-validation approach in all the models. Moreover, since we are dealing with a time series problem, it is important to use the temporal component splitting technique to preserve the chronological order of events, as random splitting could disrupt the order. We use 80% of the data for training, 10% for validation, and 10% for testing. Our regression models have an LSTM or GRU layer followed by one "relu" dense layer.

To evaluate the models' effectiveness, we use an R-squared score. Higher R-squared values indicate a slight difference between the observed data and the fitted values. A high R-squared suggests that the model's variance is similar to actual values and that they are strongly related.

4.6.3 Experimenting with How Far Can We Predict?

The first experiment in evaluating attack endgame prediction is about the relation between prediction time and obtained accuracy. The records obtained from the simulation are indexed using the time of the message received. The difference between each message is one second, which reflects the difference between every record. Every message acts as an indicator of the network state at the time when the message is sent. The network has three states: hazard, no hazard, or unknown. The goal of the experiment is to predict the occurrence of a hazard.

In our experiments, we need to choose two numbers: n (window size) and r (step size). To evaluate the model's ability to predict an event after n seconds, we take a window of n messages and then predict the message $n + 1$. After that, the window moves with the given step. For window sizes between 30 and 90, we use one step (i.e., shifting by one message); for those 91 and larger, we use ten steps. We use more steps with larger window sizes. Otherwise, the experiment will crash if small steps are applied to large windows. We also find this compatible with the nature of the problem. The farther the prediction goes, the less distant data from the actual event we will need. In other words, the older the data, the less relevant it becomes to predictions.

We use vanilla LSTM and GRU to create a baseline and to see how far we can

predict in the future and with what accuracy. As shown in Figures 4.9, 4.10, and 4.11, we can predict the occurrence of a hazard up to 3.5 minutes in advance with over 80% accuracy. We consider these auspicious results since 3.5 minutes can give enough time for countermeasure solutions to act. Given that the algorithms used here are primitives, they offer much room for improvement, as demonstrated by the third experiment.

4.6.4 Testing the Effect of Data Quality on the Prediction Process

The objective of the second experiment is to examine the impact of varying time frames on the model's predictive accuracy. We divide the overall time frame into three segments, as depicted in Figure 4.4. The subfigures showcase each time segment, with each subfigure including a different amount of data. The difference between the subfigures lies in the inclusion of messages that are further from the hazard occurrence time. Figure 4.4(a) represents the scenario in which all available data is included, Figure 4.4(b) includes less data while still capturing information beyond the hazard occurrence duration, and Figure 4.4(c) encompasses data only around the hazard occurrence time. The green highlighted areas in every subfigure represent the three different included time segments in each case, while the red highlighted areas signify hazards. Testing the model's accuracy on each time segment improves the understanding of the time-performance relationship.

In our experiments, we thoroughly evaluate each scenario depicted in Figure 4.4 and obtain insightful results. Figures 4.9, 4.10, and 4.11, provide an overview of the results. These figures show the model's predictive accuracy across the different time segments and provide insights into the impact of the time segment on the model's

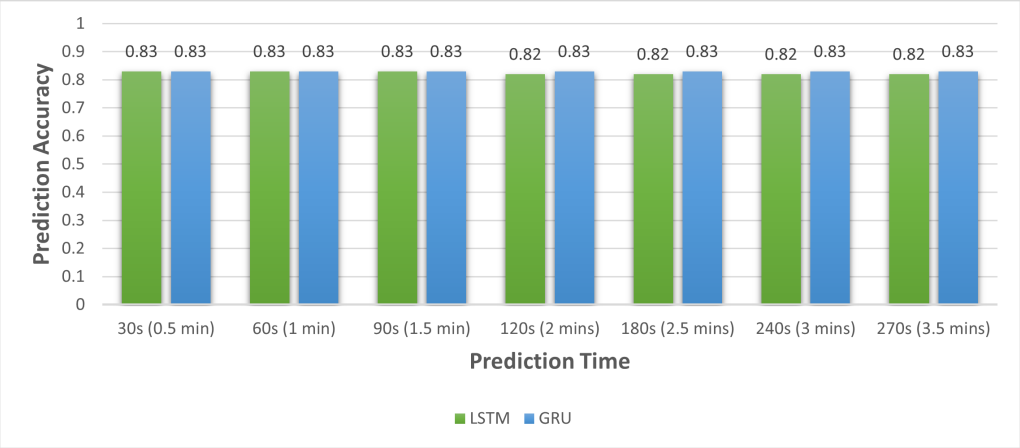


Figure 4.9: Prediction results using LSTM and GRU from 30 secs up to 3.5 mins on all available data

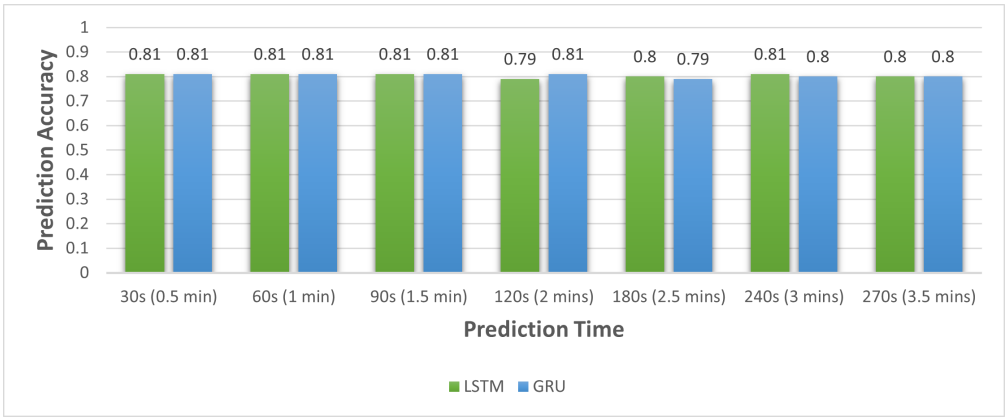


Figure 4.10: Prediction results using LSTM and GRU from 30 secs up to 3.5 mins on a smaller time window

performance. Looking at Figures 4.9, 4.10, and 4.11, we can see a decline in the overall accuracy of the results, where a number of them fall under 80%. Hence, it shows how the quality of the used data has a noticeable effect on the prediction results. The difference between the first, second, and third segments has two factors: the duration and the incorporated information. The first factor shows that not having enough time before the event (i.e., the hazard) can severely affect the prediction process.

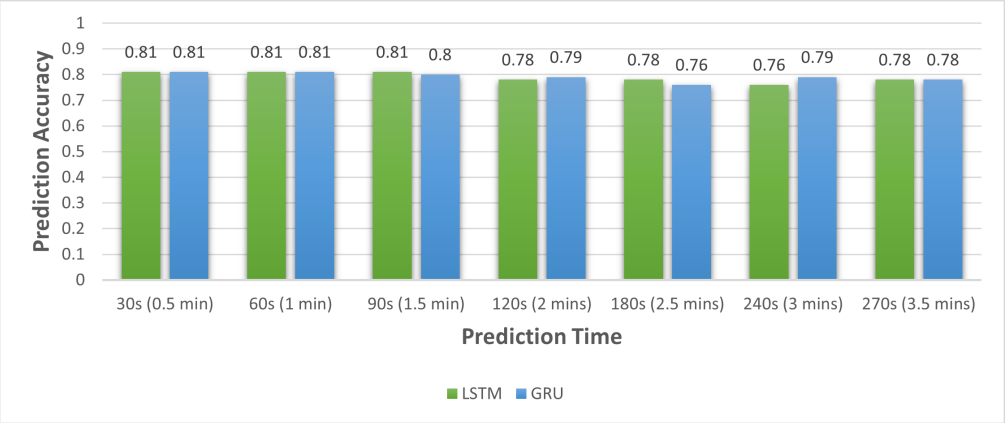


Figure 4.11: Prediction results using LSTM and GRU from 30 secs up to 3.5 mins on only hazard data

The second factor demonstrates the importance of including past recorded events that are responsible for creating the hazard. Consequently, the second factor affects the attack endgame predicting process. As shown in Figure 4.4, these events occurred when the actual attack took place, and they are important to include even though the hazard only materializes later.

4.6.5 Comparing Stacked LSTM and GRU Model Against Baseline

The purpose of this experiment is to introduce a new stack model. In this model, we stack an LSTM and a GRU model, and we compare the obtained results against previously introduced vanilla LSTM and vanilla GRU. LSTM is preferred for large data, while GRU is recommended for time-constrained problems [87]. Since VANETs undergo time constraints, and at the same time, we can accumulate a large number of data over time, we decided to stack both LSTM and GRU.

We conduct this experiment only on the last data frame, where only the data around the hazard is considered. As discussed in the second experiment, this data

frame has the lowest accuracy values. In this experiment, we demonstrate the benefit of using the introduced stacking model in improving the obtained results. Table 4.4 shows a comparison between the obtained results by all three models. We can see an increase in the R-square values in all time windows. The table shows how all the obtained results managed to break the 80% limit with the new model.

Time before hazard	LSTM	GRU	Stacked LSTM & GRU
30s (0.5 min)	0.81	0.81	0.84
60s (1 min)	0.81	0.81	0.84
90s (1.5 min)	0.81	0.8	0.84
120s (2 min)	0.78	0.79	0.83
180s (2.5 min)	0.78	0.79	0.82
240s (3 min)	0.76	0.79	0.81
270s (3.5 min)	0.78	0.78	0.81

Table 4.4: Comparison between the obtained results from LSTM, GRU, and the stacked model

4.7 Summary

This chapter presented VeReMiAP, a VeReMi-based dataset for evaluating proactive security approaches for VANETs. The need for proactive security for VANETs has become increasingly important for improving road safety and traffic efficiency. The dataset introduced in this chapter has addressed the limitations of existing datasets for VANETs by extending the VeReMi dataset using F2MD and adding CAMs and

FRA, and incorporating the effect of the attack. The inclusion of the FRA in the simulation demonstrated its potential impact on road conditions and surrounding vehicles, highlighting its significance as a major threat in VANETs. The simulated VANET environment closely mimics real-world scenarios, comprehensively representing vulnerabilities and threats in VANETs. Our simulations aimed to replicate real-world scenarios, including a low frequency of attack messages and a corresponding low frequency of hazard occurrences in the dataset. This approach provided a more realistic representation of the problem and presented opportunities for future additions to the dataset by researchers. We used the results from two RNN models as a baseline and introduced a stacked model composed of LSTM and GRU models. The final results show that the attack endgame could be predicted up to 3.5 minutes in advance with 80% accuracy. While the presented work provides insight into the potential effects of an attack, more research is needed to fully address the issue, as it is also essential to focus on developing countermeasures to mitigate the predicted effects of the attack. Thereby saving lives by avoiding the catastrophic effects of VANET cyber attacks beforehand. The dataset in this chapter will provide a helpful tool for evaluating prediction techniques and generating countermeasures for enhancing the security of VANETs.

Chapter 5

Attack Endgame Prediction in Trust Service

In this chapter, we target the second component of the proactive TMS designed for VANETs, focusing on trust service. This part highlights the crucial role of proactive strategies in managing one of VANETs' key services: security message-related services. Our approach uses a previously developed dataset to understand the potential impact of attacks on security messages and to create methods for reducing these effects [17]. Through this work, we aim to strengthen the resilience of VANETs against security threats, thus advancing trust management systems in these networks.

5.1 Introduction

As mentioned in Chapter 1.1, the security landscape for VANETs can be categorized into two distinct categories: infrastructure-centric and ad-hoc-centric (see Figure 1.1). For the infrastructure segment, Public Key Infrastructure (PKI) [88] is tasked with the authentication, issuance, and revocation of certificates for vehicles seeking to integrate into the network [89]. The ad-hoc segment utilizes trust management systems [90] to validate the integrity of information exchanged between authenticated vehicles. The use of a dual-structured approach poses significant challenges, particularly

regarding insider attackers [91] with malicious intent. Such attackers can strike at any point in the system [92]. Furthermore, the necessity for rapid and efficient security solutions is amplified by the stringent performance constraints inherent to VANETs, thus demanding robust and agile security protocols [93] that can operate effectively within these parameters [94].

Building on the preliminary focus on the ad-hoc aspect of VANETs, particularly in the context of BSMs [19] and CAMs [20], this chapter delves into the dynamics of misbehaving attacks. To address these attacks, we have utilized the VeReMi for Attack Prediction (VeReMiAP) dataset, derived from F2MD. This dataset, abundant in CAMs, allows for an exhaustive analysis of the effects of fake reporting attacks.

Distinct from other studies focusing on pre- or post-attack scenarios [95], our approach encompasses both and addresses the aftermath of attacks within VANETs. We propose a dual-faceted mitigation strategy amalgamating endgame prevention with proactive attack countermeasures. Our strategy begins with endgame prevention, where the utilization of advanced Machine Learning (ML) techniques, especially RNNs [96], plays a critical role. RNNs are designed for early detection and interception, guarding against communications that may escalate into network-wide disturbances.

Next is the attack countermeasure phase, integrating geofencing [97] and Dynamic Blacklist Management (DBM) [98]. Employing the Density-based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [99], geofencing orchestrates message flow through cluster management, strategically mitigating network hazards. Dynamic blacklist management, supported by Reputation System-based Lightweight

Message Authentication (RSMA) [100], efficiently blacklists suspect vehicles, bolstering both proactive and reactive defenses against these cyber threats. Our system underwent extensive experimental evaluation, encompassing baseline performance comparison, combined technique efficacy, sensitivity and threshold analysis, interpretation of modified reputation function in RSMA blacklisting technique, and analyzing mitigation efficacy across diverse attack densities. The results indicate significantly reduced network hazard events and improved vehicular speed recovery.

The existing literature classified various attacks within VANETs [101]. However, a notable research gap exists in quantifying and mitigating the destructive consequences of these attacks on the network. The work in this chapter endeavors to bridge this gap. Building on the previous work in Chapter 4, we extend the initial predictive approach to encompass a consideration of the cascading effects of such attacks. Subsequently, we introduce an algorithmic model designed to predict and mitigate the impact of attack endgames effectively. We refine the existing attack mitigation process by incorporating strategies to prevent an attack's final stages and counteract the attack's progression. We adopt two critical steps from the established mitigation model: endgame prevention, which employs a RNN [102] to stop the spread of harmful messages, and attack countermeasures, which focus on containing the attack and addressing its origin. Our approach includes a three-layered geofencing method using DBSCAN clustering and the convex hull algorithm to manage communication within high-traffic areas effectively. Additionally, we introduce a blacklisting technique derived from RSMA for the removal of vehicles misbehaving. We also enhance the F2MD framework, integrating these new strategies to measure the effectiveness

of attack mitigation through simulation, offering a solution to the challenge of cyber-attacks in VANETs.

The subsequent sections of the chapter are structured as follows: Section 5.2 addresses related work on attack mitigation and misbehavior detection in VANETs. Section 5.3 introduces the problem, alongside preliminaries for attack endgame mitigation, and reviews related work on attack prediction in VANETs. Section 5.4 elaborates on the proposed endgame mitigation solution, bifurcated into attack endgame prediction and countermeasure components. Section 5.5 presents the simulation results of the proposed model. Finally, Section 5.6 has concluding remarks.

5.2 Related Work

A series of studies have significantly contributed to the field of attack mitigation in both VANETs and mobile ad-hoc networks (MANETs) [103]. We extend the related work to encompass MANETs, leveraging their similarities to VANETs. Putra and Sulistyo [104] introduced a decentralized, trust-based mechanism to mitigate the impacts of Sybil attacks, which are characterized by the creation of numerous fraudulent identities to disrupt network integrity. This approach capitalizes on vehicular networks' dynamic topology, exploiting vehicles' mobility patterns and limited transmission capabilities to identify and counteract malicious entities. The method is notable for its operational independence from fixed infrastructure, presenting a significant advantage in diverse vehicular environments. However, the strategy's robustness against increasingly complex attack methodologies remains an area for potential enhancement, particularly in scenarios involving highly adaptive adversaries.

Dhanaraj et al. [105] propose a cryptographic solution for mitigating blackhole

attacks in MANETs, employing a dynamic threshold-based method. While the cryptographic methodology is robust, the work does not address the potential computational overhead introduced by the security measures. In time-sensitive vehicular network environments, such overhead could be detrimental, affecting the network's performance.

Zhuang et al. [106] worked on Emergency Message (EM) dissemination within VANETs through the innovative Time/Location-Critical (TLC) framework and Scalable Modulation and Coding (SMC) scheme, which is related to utilizing CAMs to identify and respond to fake reporting attacks in VANETs. While the TLC framework aims to optimize the delivery of genuine emergency alerts to enhance road safety and traffic efficiency, the focus on CAMs tackles the challenge of discerning and mitigating the impact of malicious false alerts that could potentially create hazardous situations. Together, these approaches highlight the critical need for robust and reliable communication systems in VANETs that not only facilitate the efficient dissemination of true emergency information but also safeguard against the detrimental effects of fabricated reports.

Several studies have concentrated on refining misbehavior detection techniques. Tsukada et al. [107] introduce a framework that utilizes collective vehicular data to pinpoint misbehavior, all while maintaining the privacy of individual nodes. The approach is commendable for considering privacy, an aspect often neglected in misbehavior detection systems. However, investigating system scalability and defense against advanced adversaries mimicking benign behaviors remains an area for further research.

Ercan et al. [108] propose fortifying the integrity of positional information within

MANETs, a critical component for maintaining operational efficacy. The work is presumably instrumental in curtailing the prevalence of location spoofing attacks. While the potential impact of this work is significant, the real-world applicability hinges on the solution's ability to function under varied network constraints and against sophisticated attack patterns that continuously evolve.

In the realm of reputation systems, Cui et al. [100] propose a framework for 5G-enabled vehicular networks. They introduced the novel reputation-based algorithm RSMA in their framework to validate messages and enhance network security. Employing elliptic curve cryptography and batch verification, the framework efficiently filters out vehicles with insufficient reputation scores, thereby reducing the presence of untrusted messages. Notably, the framework's design omits the involvement of RSUs in the authentication process, significantly reducing computational latency and security risks. Despite its effectiveness in countering common security attacks and maintaining network integrity, the framework's robustness against false positives in the reputation system remains an area for future refinement.

Kang et al. [109] tackle enhancing security in the Internet of Vehicles (IoV) by refining the delegated proof-of-stake (DPoS) mechanism with a focus on selecting trustworthy miners through a reputation-based voting system and an additional step of verification by standby miners. This dual approach aims to ensure data integrity and prevent collusion, positing increased reliability of vehicular data sharing. However, the work would benefit from a closer examination of potential challenges, particularly regarding the system's adaptability to the dynamic IoV environment and the performance of reputation-based systems across different network scenarios. While the proposed method shows promise for improving IoV security, a more thorough

critique and empirical testing are essential to validate its effectiveness and feasibility in real-world applications.

5.3 Preliminaries

In this section, we focus on the susceptibility of VANETs to fake reporting attacks. We also discuss the implications of these attacks on road safety and vehicle functionality. We investigate the objectives behind such attacks, like causing hazards and vehicle takeovers, particularly emphasizing the exploit phase where the attack's success relies on vehicle responses to deceptive messages. Further, we define "hazard" within this context. Finally, we discuss attack endgame mitigation techniques, especially geofencing in network segmentation and dynamic blacklist management in IRP.

5.3.1 Proactive Security Challenges in VANETs

The concept of proactive security in VANETs is fraught with complexities [110]. The unique constraints and inherent nature of VANETs, characterized by high mobility, dynamic topology, and stringent latency requirements [111], increase the challenge of ensuring robust security. Unlike traditional networks, VANETs must accommodate rapid changes and unpredictable patterns, making conventional security approaches ineffective.

Recognizing the importance of pre-empting security breaches [41], the work on the generated dataset in the previous chapter initially focused on finding out if we can predict the impact of attacks (i.e., attack endgame) in VANETs environments, where we successfully developed predictive models that anticipated the attack endgame.

However, prediction alone is insufficient. The true utility of these insights lies in their application towards effective attack mitigation. The transition from understanding the potential impact of an attack to actively countering it is key to improving security in VANETs.

Multifaceted challenges hinder the implementation of proactive and prediction-based security solutions. Firstly, the need for real-time data analysis and threat intelligence strains the limited computational resources inherent to vehicular systems. This limitation poses a significant barrier to executing complex predictive algorithms. In addition, a security system must consider the delicate balance between minimizing false positives to avoid desensitization to security alerts and reducing false negatives to ensure no genuine threats go undetected. Moreover, the necessity for real-time response mechanisms, integral to mitigating potential attacks promptly, underscores the importance of speed in such systems. Integrating these advanced security measures with existing protocols without introducing new vulnerabilities remains a concern. Lastly, the reliance on collaborative security approaches, which necessitates a harmonized balance between efficient inter-vehicle communication and stringent privacy safeguards, adds another layer of complexity to proactive VANET security.

The next subsections discuss various mitigation techniques and analyze their applicability and effectiveness in the VANETs. We discuss how these adapted techniques can be employed to fortify VANETs against the threats to which they are susceptible.

5.3.2 VANET Attack Life Cycle

We introduce a proposed version of the attack life cycle in VANETs to describe where proactive security can be applied through the cycle. Figure 5.1 compares the phases

of the cyber attack life cycle against the VANET attack life cycle. The figure also specifies what phases belong to proactive security and what should belong to reactive security.

As shown in Figure 5.1, reconnaissance is the first phase in the cyber attack life cycle, and it is the same in VANETs. In the reconnaissance step, the attacker develops a target for their attack. The second step, weaponizing, is when the attacker creates the payload of the attack. In VANETs, however, this is where the attacker begins to create a false message (either a BSM or traffic warning message). The third step, deliver, is where the vulnerability in the system is weaponized. In VANETs, this is where the attacker exchanges false messages by broadcasting them to the surrounding vehicles. The fourth step, exploit, is where the weaponized payload is executed. However, in VANETs, this is where the vehicles start to act upon the received message. The remaining steps, control, execute, and maintain, are more related to a typical cyber attack than VANETs. In these steps, the attackers try to escalate their privileges and maintain their presence in the system by creating back doors. Whereas in VANETs, we have what are called attack effects.

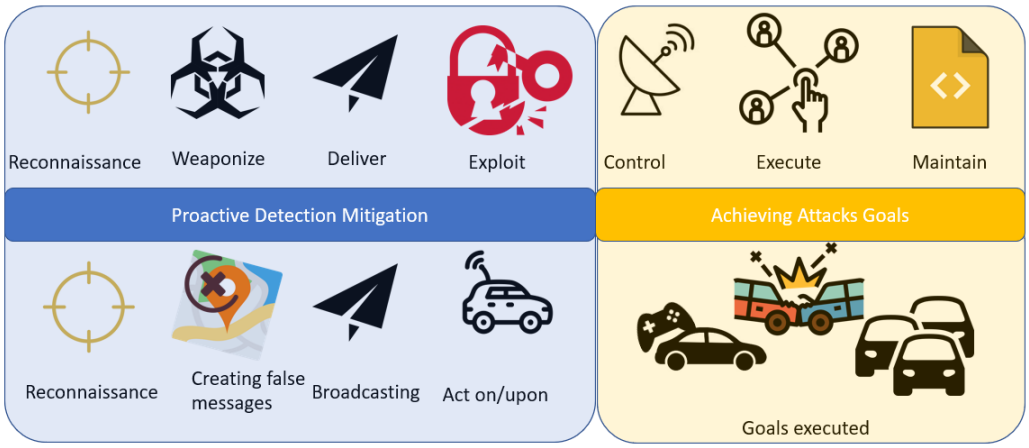


Figure 5.1: Introducing attack life cycle in VANET

The attack goals are the attack's impact on the vehicles that acted on the received false messages. Attack goals can create a hazard, an accident, or take over control of the vehicle [112]. To achieve the attack goals, the vehicle has to receive the false message and behave accordingly. In VANETs, receiving the attacker payload (i.e., the false message) does not imply that the payload fulfilled its purpose. We can only say that the exploit phase succeeded when the vehicles started to act upon the incorrect information.

5.3.3 Attack Endgame

Various attack endgames have been identified in VANETs, including data theft and privacy breaches [113], network disruption, denial of service (DoS) [114], and resource drainage [115]. However, there is a notable lack of research specifically targeting the creation of road hazards [116] despite its significant implications. This gap is particularly striking given that fake road hazard creation could affect human lives, arguably more so than other types of attacks [117]. Recognizing the profound impact of road hazards on both safety and traffic efficiency, We aim to contribute to mitigating the destructive consequences of such attacks, focusing on strategies that predict and prevent scenarios that could lead to serious accidents.

In the context of our research within VANETs, a hazard is defined with two specific quantitative parameters: the speed of vehicles and the number of affected vehicles. A situation is classified as a hazard when the number of vehicles impacted surpasses a predetermined limit, concurrently coupled with a reduced speed below a defined minimum. This definition objectively recognizes a hazardous condition on the road. The rationale behind this definition lies in the correlation between high vehicle density

and reduced speed, which often indicates potential risks such as traffic congestion or an obstructive event. By quantifying a hazard regarding vehicle count and speed, we can accurately identify and respond to situations that threaten road safety. This approach allows for a data-driven assessment of traffic conditions, facilitating timely and effective interventions to mitigate potential dangers in vehicular networks.

5.3.4 Attack Mitigation

Figure 5.2 delineates the methodology for attack mitigation, comprising various steps, each with its specific sub-steps, to ensure effective outcomes: (1) Risk Assessment, which entails identifying and evaluating potential security threats and vulnerabilities within the system, along with assessing their probable impact; (2) Attack Prevention, a proactive stage focused on implementing defensive measures such as firewalls, intrusion detection systems, and secure system configurations aimed at averting the onset of attacks; (3) Attack Countermeasure, a crucial phase involving activating specific response mechanisms to address and neutralize attacks in progress. This phase is characterized by isolating affected systems, constraining the extent of the attack, and finally removing its causative elements; (4) Recovery, where the emphasis is reinstating normal operational states and rectifying any damage sustained from the attack; (5) Review and Improvement, are imperative for the evolution of the attack mitigation strategy. This involves a rigorous post-incident analysis to identify enhancement opportunities, fortifying the system against similar threats in the future.

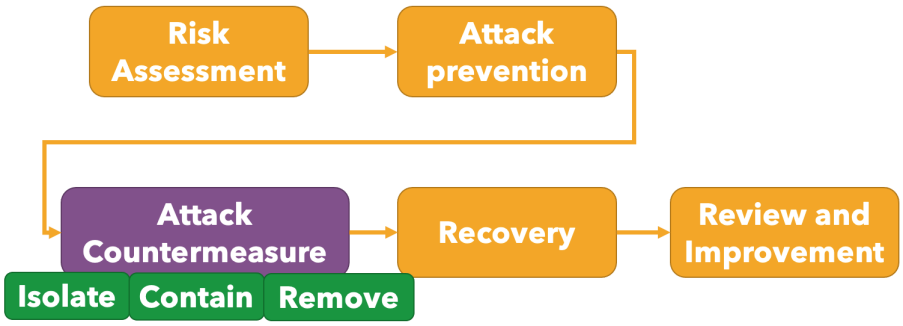


Figure 5.2: Attack mitigation process: A generic representation

5.3.5 Attack Countermeasure

In Chapter 4, we made significant strides in attack prevention within VANETs, developing a novel methodology for predicting and preventing the attack endgame. Building on this foundation, the work in this chapter focuses on devising an effective attack countermeasure strategy to minimize the impact of the attack endgame in VANET environments. As illustrated in Figure 5.2, the attack countermeasure comprises three sub-steps: isolate, contain, and remove.

Due to the nature of VANETs, treating vehicles as regular network nodes and severing their network connections is not feasible without unequivocal evidence of compromise. Consequently, this work addresses containment and removal.

5.3.6 Attack Endgame Mitigation Techniques

In order to apply attack mitigation, we had to search different proposed attack mitigation techniques. Then, we identified the relevant ones for use in the containment and removal stages of the attack mitigation process. These techniques strengthen systems against malicious threats, each offering a distinct defense aspect in this area and include:

- Incident Response Planning (IRP) [118],
- Network segmentation [119],
- Firewalls [120],
- Intrusion Detection Systems (IDS) [121].

These techniques in VANETs build a strong defense against the attack endgames. We focused on identifying the most effective and key concepts. Incident response planning and network segmentation are particularly significant, and they are described in Section 5.3.7 and Section 5.3.8, respectively. Network segmentation is a containment-related technique that helps create a divided network structure, improving security by separating different network parts. This separation helps stop attacks from spreading too widely. Incident response planning is a removal-related technique that sets up a clear and quick action plan, allowing for fast and effective responses to emerging attack situations, thereby reducing potential harm and disruption.

5.3.7 Incident Response Planning

IRP plays a key role in managing a well-organized response to security breaches. Some parts of IRP are particularly effective within the scope of Attack Endgame Mitigation in VANETs. Containment isolates affected areas of the network or specific devices, which helps stop the attack's spread and reduces its overall damage. It is critical to distinguish that this containment step in IRP marked differs from the 'Contain' step observed in the attack mitigation process. The containment phase in IRP primarily focuses on immediate response actions, specifically aimed at cordoning off the affected

areas or devices within the network to halt the attack's spread. In contrast, the Contain step in the attack mitigation process encompasses a broader strategy involving isolation and the implementing measures to prevent further infiltration or damage by the attack. While both are critical in managing network security threats, the containment phase in IRP is more reactive and immediate. In contrast, the Contain step in attack mitigation involves a proactive approach to network defense. Analysis involves gathering and examining important information like system logs and network traffic. This analysis helps create targeted response strategies for VANETs, ensuring that actions taken match the specific nature of the attack. Finally, recovery focuses on returning the network to normal and fixing any affected functions. This step is key in dealing with the lasting effects of the attack. We have adopted dynamic blacklist management as a strategic mechanism, recognizing its alignment with IRP's foundational steps and its efficacy in preemptively mitigating network threats.

Dynamic Blacklist Management

In the orchestration of VANET mitigation methods, dynamic blacklist management encompasses most of the IRP steps, focusing on stages such as monitoring, detecting suspicious activities, blacklisting vehicles' pseudonyms, and subsequently lifting the blacklist when deemed appropriate. Strategies such as machine learning-based prediction, rule-based prediction, Statistical prediction, and Hybrid prediction are typically utilized in dynamic blacklist management. A hybrid prediction method is adopted, combining machine learning and rule-based strategies. This synergy of methodologies leverages the extensive detection capabilities of machine learning, coupled with the rapid and precise identification provided by rule-based approaches.

Machine Learning and Rule-Based Hybrid Approach: The ML component employs LSTM networks to capture temporal dependencies, establishing a reliable detection system for identifying malicious activities. The rule-based component employs both threshold-based and consensus-based predictions within this integrated approach. Threshold-based prediction relies on specific, quantifiable metrics, where exceeding certain thresholds initiates the blacklisting process. Conversely, consensus-based prediction encourages cooperative behavior, where collective assessments from vehicles aid in pinpointing and blacklisting malicious entities.

5.3.8 Network Segmentation

Various segmentation strategies can be adeptly applied to fortify VANETs against potential adversities. Cluster-based segmentation facilitates the grouping of vehicles into clusters based on proximity, optimizing data flow control and access within closely located vehicular nodes. Geographical segmentation aligns network segments with geographical demarcations, fostering efficient communication and bolstering security within specified geographical confines. Rule-based segmentation divides network segments according to the functional roles of vehicles, such as prioritizing and securing communications for emergency vehicles. Hybrid segmentation amalgamates multiple segmentation techniques, harnessing their collective strengths. Each segmentation technique brings a dimension of defense, collaboratively contributing to a fortified network architecture resilient against the diverse challenges intrinsic to VANETs.

Geofencing in Network Segmentation

Geofencing emerges as a promising technique in the network segmentation of VANETs, playing a pivotal role in mitigating the propagation of malicious activities, and it is particularly those about disseminating deceptive accident CAM messages. Below, we detail the steps involved in implementing geofencing in this setting:

1. **Defining Geofence Boundaries:** The first step in geofencing is establishing the geofence's boundaries. This involves choosing the area surrounding malicious vehicles and the affected zone. Technologies like GPS and Geographic Information Systems (GIS) are key in accurately setting these boundaries. The boundaries are further refined by considering the transmission range of technologies such as DSRC and Cellular-Vehicle-to-Everything (C-V2X), ensuring complete coverage of the affected area.

2. **Determining Trigger Events:** Identifying specific trigger events, such as the transmission of false accident reports or hazards, is essential. Security systems like Intrusion Detection and Prevention Systems (IDPS), enhanced with machine learning algorithms, are instrumental in detecting unusual communication patterns and flagging them for intervention.

3. **Action or Event Configuration:** Once a harmful communication within the geofenced area is detected, measures are implemented to mitigate its impact. This includes regulating vehicle speeds via V2V communications or dropping malicious messages, thereby preventing a hazardous event.

5.4 Proposed Scheme

In Figure 5.4, we present an extended version of the attack mitigation process depicted in Figure 5.2, tailored specifically to the context of VANETs and attack endgame. This adaptation aligns with the inherent characteristics and operational dynamics of VANETs. We provide a detailed mapping and explanation of how each step of the attack mitigation process is conceptualized and executed within the VANET environment, focusing on the attack endgame scenario; the steps are as follows:

1. **Endgame Assessment:** At this initial stage, the system evaluates exchanged messages to identify the potential attack endgame and assesses the likelihood of such an event occurring.

2. **Endgame Prevention:** Following the assessment, the next step involves proactive measures to stop the messages causing the attack, effectively preventing the endgame from manifesting.

3. **Attack Endgame Countermeasure:** Should an attack transpire, this phase activates a multi-step countermeasure approach, incorporating the following substeps:

    - *Isolate:* Segregating affected system components to prevent the proliferation of the attack's impact.
    - *Contain:* Implementing measures to contain the impact of the attack endgame and prevent further compromise.
    - *Remove:* Executing actions to remove misbehaving vehicles from the network to halt the attack.

4. **Recover:** Mitigating the incident's impact by efficiently managing traffic flow and vehicle speeds through dynamic traffic rerouting and speed regulation. Controlled speed limits are enforced within and surrounding the geofenced area.

5. **Review and Improve:** Conducting a review of the mitigation process to identify areas for improvement, enhancing the system's resilience against future attacks.

Attack endgame mitigation in VANETs primarily constitutes attack endgame prevention and countermeasures. Our approach focuses on specific steps that exhibit potential for targeted mitigation of the attack endgame in VANETs.

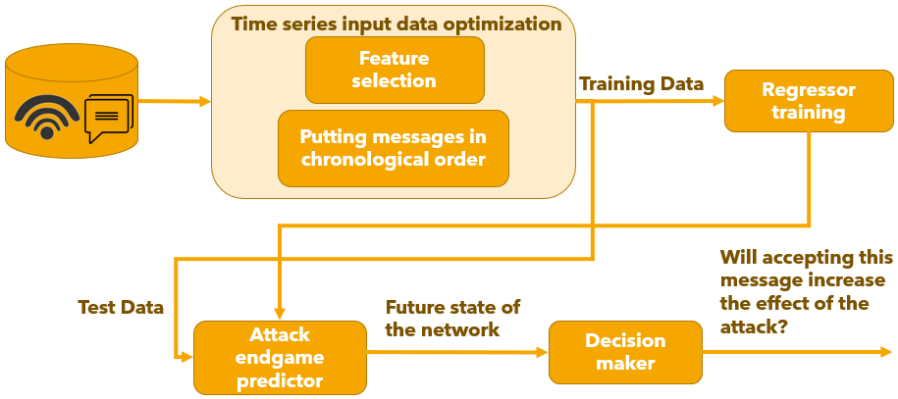


Figure 5.3: Predicting attack endgames in VANETs

5.4.1 Attack Endgame Prevention

Figure 5.3 illustrates a workflow designed for predicting the attack endgame in VANETs, starting with the input of exchanged BSMs and CAMs. Initially, these messages undergo a time series data preparation process, including feature selection to identify relevant data points and arranging the messages chronologically to preserve the sequence of events. The optimized data is split into two streams: training data for

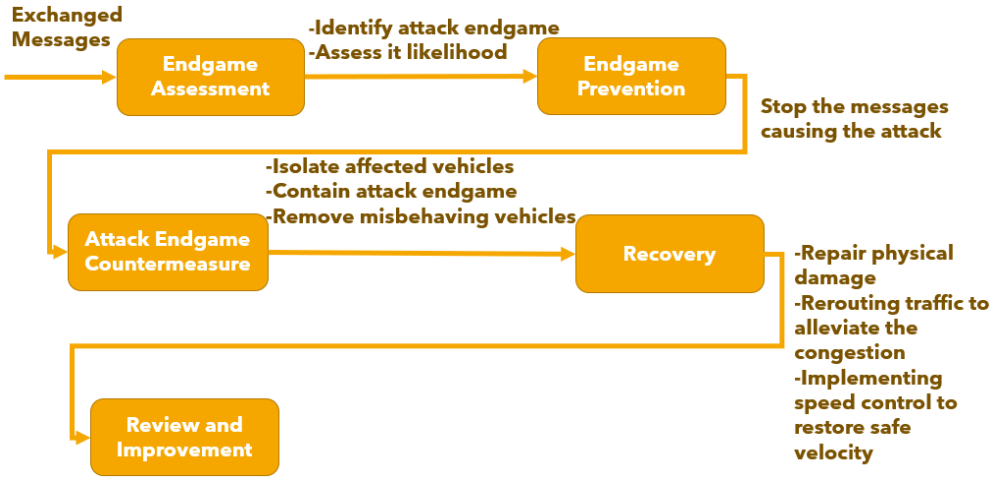


Figure 5.4: Steps for proactive attack endgame mitigation in VANETs

building the predictive model and test data for evaluating the model performance. In the training phase of the predictive model, a regressor is trained on the selected features, fine-tuning the model to predict the future states of the network. For the testing phase, the model, referred to as the attack endgame predictor, takes in new data to predict the endgame of ongoing attacks. Finally, the predictions are fed into a decision-maker component that assesses whether accepting a message will exacerbate the effect of an attack, leading to a decision on how to act on the incoming messages to mitigate potential threats.

Next, we detail the Recurrent Neural Network (RNN) model implemented in the attack endgame predictor and the corresponding steps and configurations integral to its operation.

Recurrent Neural Network Architecture

The incorporated RNN architecture is designed with a memory component, enabling the storage of information from prior inputs to influence future inputs and outputs.

Given a sequence of inputs $(x_1, x_2, \ldots, x_t)$, the hidden state h_t of the LSTM is computed as:

$$h_t = f_W(h_{t-1}, x_t) \tag{5.1}$$

where f_W is a function parameterized by weights W, representing the LSTM's cell.

Prediction Strategy

We aim to predict the impact of a false traffic alert disseminated by malicious vehicles. The predicted outcome is the hazard resulting from vehicles acting upon malicious information. An LSTM model is used as a baseline to understand the extent to which prediction is achievable in foreseeing future impacts.

Window and Step Sizes

Two significant parameters in the proposed methodology are the window size n and step size r. A window of n messages is considered, and the model predicts the $n + 1$ message. The window then shifts by a step size r, and the process repeats. Mathematically, for a given sequence S, the windowing process can be represented as:

$$W_i = (m_i, m_{i+1}, \ldots, m_{i+n-1}) \rightarrow m_{i+n} \tag{5.2}$$

where W_i is the window of messages, and m represents individual messages in the sequence.

Cross-Validation

A cross-validation approach is utilized, incorporating a Train-Validation-Test split technique. Given the temporal nature of the data, temporal component splitting is adopted to maintain the chronological integrity of events. The dataset is divided as 80% for training, 10% for validation, and 10% for testing.

Algorithm 5.1 RNN Model Architecture for Attack Endgame Prediction

1: Input: Sequence of messages $(m_1, m_2, \ldots, m_t)$
2: LSTM Layer: $h_t = f_W(h_{t-1}, x_t)$
3: Dense Layer: $y = \text{ReLU}(W_h h_t + b)$
4: Output: Predicted impact y =0

Model Architecture

Algorithm 5.1 presents an RNN model architecture that is specifically designed for the prediction of attack endgames. The process initiates with the input of a time-ordered sequence of messages, symbolized as $(m_1, m_2, \ldots, m_t)$. The LSTM layer, a distinctive form of RNN known for its proficiency in managing sequential data, processes each message. It updates its hidden state h_t at every timestep, influenced by the present input x_t and the antecedent hidden state h_{t-1}, following a function f_W parameterized by the set of weights W.

Subsequent to the LSTM layer is a Dense layer, a conventional neural network layer characterized by its fully connected nature, which takes the LSTM's output h_t and applies a rectified linear unit (ReLU) activation. This is mathematically represented as $y = \text{ReLU}(W_h h_t + b)$, where W_h denotes the weights of the Dense layer and b signifies the bias term.

5.4.2 Attack Endgame Countermeasure

Prior discussions in Sections 5.3.7 and 5.3.8, focused on refining the latter two substeps within the attack endgame countermeasure phase, specifically containment and removal. To address the containment step, we employed geofencing techniques, while for the removal step, we adopted dynamic blacklist management. The following section provides a detailed explanation of the methods used in each step.

Geofencing

The geofencing mechanism in the proposed system unfolds in three key steps:

1. Identify High-Density Regions: Analyze and pinpoint areas with high vehicular concentration prone to hazards.

2. Create Geofences: Set up virtual boundaries around these areas to regulate vehicle communications.

3. Curate Communications: Filter messages from within these zones to prevent hazard spread.

Geofencing Mechanism: Geofencing is implemented via the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [99] algorithm to manage and control the flow of communication within congested areas. By establishing geofenced clusters, the system can isolate and restrict the communication of vehicles within these high-density zones (e.g., clusters). Messages originating from vehicles inside these clusters are selectively dropped, thereby curtailing the spread of harmful information and reducing overall network congestion. Geofenced areas can improve

cooperation and information sharing among vehicles inside the cluster while protecting the vehicles outside, enabling a swift collective response to hazards [122].

Rationale for DBSCAN in geofencing: DBSCAN excels as a clustering algorithm by identifying densely packed data points as clusters, differentiated by areas of lower density, marking a shift from centroid-based clustering methods like K-means to a model capable of uncovering clusters of arbitrary shapes and sizes. This feature is invaluable in real-world scenarios, including vehicular traffic patterns, where data often assume complex, non-convex forms. DBSCAN's renowned proficiency in spatial data analysis aligns well with the dynamic and heterogeneous nature of these patterns, making it especially suitable for applications within such contexts.

The algorithm operates on two primary parameters: *minPts*, the minimum number of points required to form a dense region, which allows the algorithm to be sensitive to the scale of clustering, and ε (epsilon), the maximum distance within which points are considered to be neighbors. This distance parameter ε is pivotal, as it defines the spatial reach of clustering, ensuring that only genuinely adjacent data points are amalgamated into a single cluster.

The Euclidean distance metric serves as the backbone for the clustering process, given its geometric congruence with the physical dimensions of vehicular proximity. This traditional measure of distance is particularly apposite for encapsulating the direct vehicular interactions within a network characterized by Euclidean spaces, thus facilitating the formation of geofences that are both logical in structure and practical for enforcement.

Moreover, DBSCAN's inherent resistance to outliers fortifies the geofencing approach against anomalous data points, which could represent adversarial entities or

system faults. Consequently, DBSCAN, with its Euclidean distance paradigm, is the best choice for establishing robust geofencing mechanisms in the VANET security framework. These mechanisms enable the filtration of communications from identified high-risk zones and reinforce the network's resilience to opportunistic and orchestrated threats.

Figure 5.5: Generated traffic clusters on road network over time

The convex hull algorithm is integral to constructing efficient geofencing solutions in-vehicle communication networks. It encapsulates all the necessary data points within the simplest possible polygon, ensuring comprehensive coverage. The convex hull algorithm excels in precision and computational efficiency and is particularly suited for dynamic vehicular environments. By implementing the convex hull algorithm, we can create geofences that are accurate and responsive to real-time changes, establishing a secure and effective management system for vehicle clusters. Figure 5.5 shows the created clusters over time in different areas on the map. As seen on the map, a single or multiple clusters might form at some point.

Algorithm 5.2 Geofencing Using DBSCAN Clustering

Inputs:
df_Test: DataFrame with 'x_pos', 'y_pos'
eps: Max distance for neighborhood consideration
min_samples: Min samples for a core point
threshold: Min vehicles for a high-density cluster
near_distance: Max distance to be near high-density

procedure DBSCAN_Geofencing(*df_Test*, *eps*, *min_samples*, *threshold*, *near_distance*)
    Initialize *dbscan* with *eps*, *min_samples*
    Assign clusters to *df_Test* using *dbscan*
    Extract high-density clusters
    Mark vehicles in/out of high-density clusters
    for all *vehicles* not in high-density clusters **do**
        if *vehicle* on a road that has high-density clusters **then**
            Calculate *distance* from *vehicle* to all vehicles in high-density cluster
            if *distance* $\leq$ *near_distance* **then**
                Mark *vehicle* as 'Near Hazard'
            end if
        end if
    end for
    for all *cluster_id* in *high_density_clusters* **do**
        Calculate Convex Hull for the cluster
    end for
end procedure

Geofencing Implementation Steps

1. **Identify Clusters:** Use DBSCAN for finding high-density areas in the BSMs and CAMs dataset as potential clusters.

2. **Construct Geofence Boundaries:** Apply a Convex Hull algorithm to outline geofence boundaries for each cluster, forming a minimal enclosing polygon.

3. **Implementing the Algorithm:** Utilize well-established methods (e.g., Graham's Scan) for Convex Hull implementation, sorting points by polar angle and

constructing the hull.

4. **Execute Convex Hull Algorithm:** Perform the Convex Hull algorithm on DBSCAN's cluster points to create the polygonal geofence.

Algorithm 5.2 presents a structured method to apply geofencing in VANETs using the DBSCAN algorithm. The algorithm starts by setting key parameters such as the positions of vehicles (x_pos and y_pos), the maximum distance for considering two vehicles as neighbors (eps), the minimum number of neighbors for a vehicle to be a core part of a cluster (min_samples), and the criteria for identifying high-density vehicle clusters (threshold and near_distance) (lines 1 - 6). Following this setup, the algorithm uses DBSCAN to group vehicles based on proximity and identifies vehicles that are in or near high-density clusters, which can be potential hazards. First, the algorithm marks vehicles that are in or out of high-density clusters (line 11). It then iterates over all vehicles that are not already in these clusters. For each of these vehicles, the algorithm checks if the vehicle is on a road that contains high-density clusters. If this condition is met, it calculates the distance from the vehicle to all vehicles in the high-density cluster (line 14). If the distance is less than or equal to a predefined threshold (near_distance), the vehicle is marked as 'near hazard' (line 16). This process helps in identifying and managing potential risks by monitoring vehicles' proximity to high-density areas. Additionally, the algorithm outlines the high-density areas by calculating Convex Hulls, offering a visual boundary for each cluster (line 21).

Algorithm 5.3 Dynamic Blacklist Management: Reputation Score Calculation

Inputs:
threshold_feedback (Range: 10 to 90)
effectScore_low (Range: 0 to 1)
effectScore_mid (Range: 0 to 1)
effectScore_high (Range: 0 to 1)
Ensure: effectScore_low + effectScore_mid + effectScore_high = 1
procedure REPUTATIONSCORE($targetVehicle, \alpha, \beta, \gamma$)
    $D \leftarrow$ effectDegree($targetVehicle$)
    $E \leftarrow$ objectiveEvaluation($targetVehicle$)
    $H \leftarrow$ historicalReputation($targetVehicle$)
    $RS\_new \leftarrow \alpha \cdot H + \beta \cdot E + \gamma \cdot D$
    return RS_new
end procedure
procedure EFFECTDEGREE($targetVehicle$)
    feedbackPct $\leftarrow$ feedbackPctDic.get($targetVehicle$, None)
    if feedbackPct $\geq$ threshold_feedback **then**
        effectScore $\leftarrow$ effectScore_high
    else if threshold_feedback/2 $\leq$ feedbackPct $<$ threshold_feedback **then**
        effectScore $\leftarrow$ effectScore_mid
    else
        effectScore $\leftarrow$ effectScore_low
    end if
    $N \leftarrow$ feedbackVehicleDic.get($targetVehicle$, {}).get($'count'$, 0)
    return effectScore $\cdot N$
end procedure
procedure OBJECTIVEEVALUATION($targetVehicle$)
    feedbackVehicles $\leftarrow$ feedbackVehicleDic.get($targetVehicle$, {}).get($'vehicles'$, [])
    reputationScoreList $\leftarrow$ [reputationScoreDic.get($vehicle$, 0) for vehicle in feedbackVehicles]
    Calculate E based on reputationScoreList
    return E
end procedure
procedure HISTORICALREPUTATION($targetVehicle$)
    return reputationScoreDic.get($targetVehicle$)
end procedure

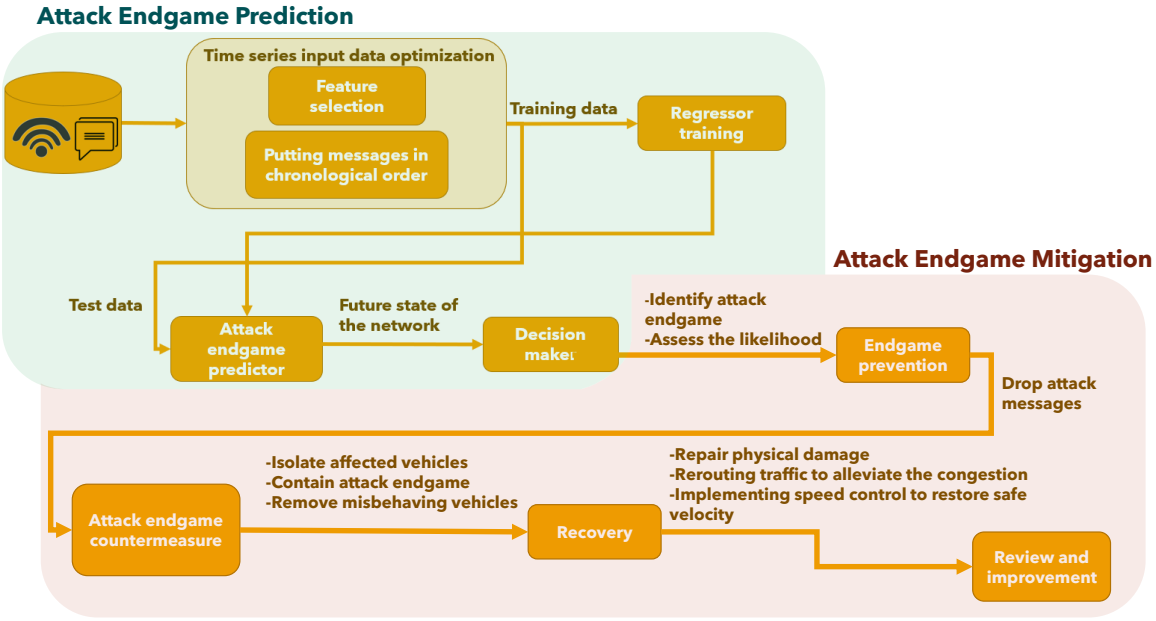


Figure 5.6: Integration of attack endgame prediction and mitigation in VANETs

Dynamic Blacklist Management

Dynamic blacklist management is a critical component in implementing the removal step. Adopting the reputation algorithm, intricately woven into the reputation RSMA algorithm, is a pivotal cornerstone in this implementation.

Rationale for RSMA in Dynamic Blacklist Management: A confluence of strategic advantages underscores the selection of the RSMA algorithm. Primarily, its intrinsic compatibility with CAMs and the centrality of authority mechanisms bolster its applicability. Furthermore, the algorithm considers the malicious message's consequential effects. Augmented by its advanced developmental maturity and enhanced scholarly recognition, the reputation algorithm of RSMA emerges as an astute choice for dynamic blacklist management.

Modifications to the RSMA Algorithm: Several modifications have been

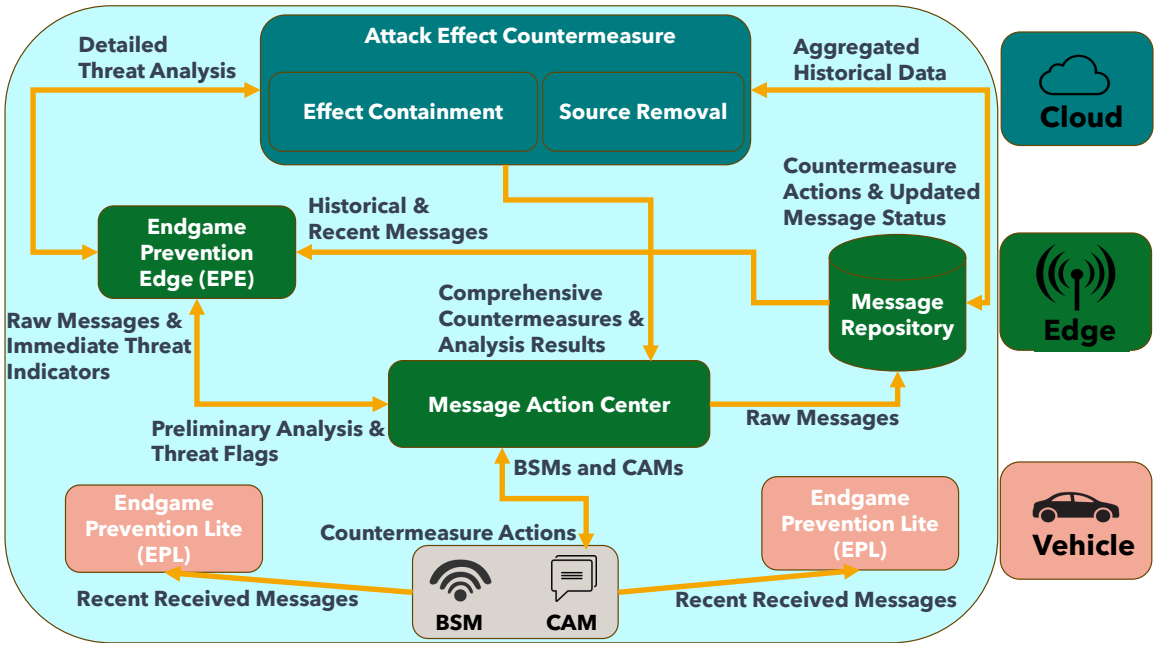


Figure 5.7: VANET security architecture for attack endgame prediction and mitigation

integrated into the RSMA algorithm to enhance its adaptability and efficacy. Eliminating the constraint of time windows and introducing a novel definition of "Message Effect" have been instrumental modifications. Additionally, feedback participation has been democratized, allowing a broader range of vehicular entities to contribute, thereby enriching the algorithm's evaluative depth.

Steps of the RSMA Algorithm:

1. *Effect Degree:* This step quantifies the impact, multiplying the number of participants by the message effect.

2. *Objective Evaluation:* An aggregation stage where vehicles within reputation score intervals are weighted and collated.

3. *Historical Reputation:* This step involves leveraging historical data in the evaluative process.

4. *Final Reputation:* Culmination of the previous steps to ascertain the final reputation score.

Through the implementation and adaptation of the RSMA algorithm within the DBM framework, a robust and resilient strategy is cultivated, driving the effectiveness of IRP in VANETs.

Algorithm 5.3 primarily focuses on computing the reputation score of vehicles in VANETs, which is pivotal for efficient DBM. The algorithm calculates a target vehicle's reputation score by combining three factors: the effect degree, objective evaluation, and historical reputation. The reputation score procedure (lines 8-12) combines the three components — effect degree (D), objective evaluation (E), and historical reputation (H) — using given weights (α, β, γ). The final reputation score (RS_new) is calculated and returned. The effect degree procedure (lines 16-24) determines the effect score for a target vehicle based on its feedback percentage. Depending on the feedback percentage, the vehicle is assigned an effect score (high, mid, or low). The final effect degree is calculated by multiplying this score by the number of feedbacks and then returning it. The objective evaluation procedure (lines 27-30) retrieves the feedback vehicles for the target vehicle and calculates an objective evaluation score (E) based on the reputation scores of these feedback vehicles. The score is then returned. The historical reputation procedure (line 33) returns the historical reputation score of the target vehicle from the reputation score dictionary.

5.4.3 Proactive and Retroactive Mitigation of Attack Endgames

In Figures 5.6 and 5.7, an illustration of the proposed system is presented. These figures delineate the integration of prediction and mitigation processes, demonstrating their collaborative function in curtailing the ramifications of the attack's endgame.

Figure 5.6 delineates the proposed approach to mitigating the attack endgame within vehicular networks. The figure shows the amalgamation of attack endgame prediction with subsequent mitigation techniques, illustrating a multi-faceted defense mechanism. It underscores the importance of striving to preempt attacks through prediction and prevention and ensuring that, should prevention efforts fail, the system is equipped to mitigate the repercussions based on the insights provided by the prediction models. The figure shows the process from initial data optimization and feature selection through to using regression models for prediction, leading to decisive action. Should an attack transpire, the model facilitates containment and removal of the threat, followed by a recovery phase to restore normalcy. The process culminates in a review phase to refine the mitigation strategy, thus enhancing the system's resilience against future attacks.

Figure 5.7 presents a detailed view of the main components of the system, how they are interconnected, and where they are located within the network. Our system includes Endgame Prevention Lite (EPL), a simplified version of the endgame prevention system. EPL is integrated into vehicles to enable early detection and rapid response mechanisms. The BSMs and CAMs processed by the EPL are transmitted to the Message Action Center simultaneously. This center subsequently forwards these messages to both Endgame Prevention Edge (EPE) and Message Repository while also orchestrating the dissemination of countermeasure directives back to the

vehicles.

The endgame prevention model operating at the edge possesses enhanced data access and extended processing time, allowing for a more thorough analysis. The refined data obtained are then relayed to the Attack Countermeasure System, tasked with addressing the anticipated attack endgames. This system also communicates its findings back to the Message Action Center, which plays a pivotal role in determining whether to discard messages identified as part of the attack.

Furthermore, the Message Repository continuously accumulates messages, contributing additional data to the Attack Effect Countermeasure for use in the mitigation process. It also retains updated statistics on message interactions and the specific countermeasures employed for each message, serving as a vital resource for future predictive analysis.

In the final stage of the process, the Attack Effect Countermeasure, upon reviewing the insights gleaned from both the EPE and the Message Repository, executes strategic actions such as geofencing or blacklisting, targeting specific areas and vehicles. These decisions are then conveyed back to the Message Action Center, completing the system's comprehensive approach to threat mitigation.

5.4.4 Adopted Dataset and Communication Technology

The work presented in this chapter utilizes the VeReMiAP dataset [18], a detailed collection of vehicular communication data accessible from GitHub, 2 gigabytes in size, and encoded in JSON format. This dataset chronicles the sequence of messages exchanged between vehicles, with BSMs and CAMs logged at one-second intervals. Each entry, representing either a BSM or CAM, includes standard data fields, with

two additional fields for our analysis. The fake reporting attack field signifies messages that falsely report incidents, while the attack effect field records the timing of resultant hazard events, independent of the timestamps of the attack messages.

Simulation Scenario: LuST scenario [83] was selected for its high-fidelity representation of a VANET environment, leveraging real-world vehicular and traffic data. This scenario includes various urban elements such as intersections, roundabouts, and highways, providing a rich context for VANET research.

Attack Scenario: We focus on the fake reporting attack for the attack model. In this scenario, malicious vehicles disseminate false information regarding their status or road conditions. Such deceptive communications can lead vehicles to take unnecessary defensive actions. Fake reporting attacks are particularly dangerous as they directly impact road safety and create hazardous traffic conditions.

Communication Technology: Our experiments utilized the DSRC technology for communication. Despite the evolution of cellular technologies that offer extensive coverage and cater to non-emergency functionalities such as entertainment systems, DSRC remains prominent in the VANET security landscape [25]. This is attributed to its steadfast reliability, the ability for direct communication bypassing the need for cellular infrastructure, and the dedicated spectrum allocation crucial for the necessities of real-time, safety-oriented vehicular communication.

5.5 Performance Evaluation

We present a series of five experiments, each designed to evaluate and validate different aspects of the proposed mitigation strategies within VANETs. The five experiments and their purpose are as follows,

1. Baseline comparisons serve as a foundational assessment, establishing a benchmark against which the effectiveness of the proposed techniques can be measured.

2. Combinatorial approaches delve into the synergistic effects of combining various mitigation methods, exploring how these integrations enhance overall system resilience.

3. Sensitivity and threshold experiments are then conducted to understand the robustness of the system under varying operational parameters, offering insights into optimal configurations.

4. Interpretation of the modified reputation function in RSMA blacklisting technique and an in-depth analysis of the tailored approach to DBM.

5. Analyzing mitigation efficacy across diverse attack densities investigates how the system performs under different levels of adversarial pressure, assessing its adaptability and effectiveness in varied attack scenarios.

5.5.1 Baseline Comparisons

In this experiment, we evaluate the individual effectiveness of each mitigation technique in reducing hazard events within VANETs. This analysis establishes a foundational understanding of each technique's impact, which will be used in a later analysis to determine their combined effects.

As depicted in Figures 5.8 and 5.9, the endgame prevention technique exhibited the most substantial decrease in hazard events, reducing the count to 24,361, a remarkable 37% reduction. This significant decrease highlights the robustness of employing

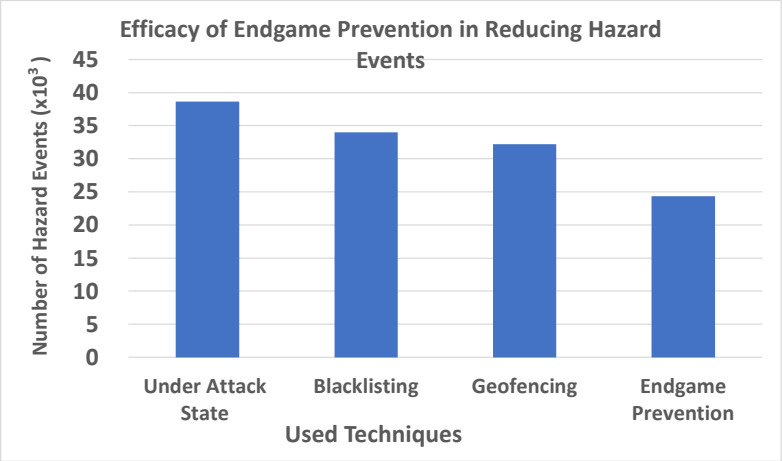


Figure 5.8: Impact of endgame prevention technique on hazard event reduction

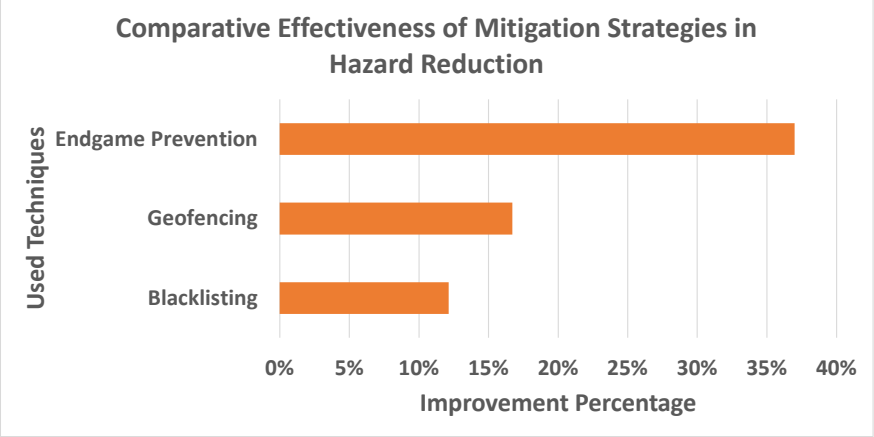


Figure 5.9: Comparative analysis of hazard event reduction percentages by mitigation technique

machine learning models for the predictive prevention of hazardous scenarios.

Upon analysis, it was observed that in the state under attack scenario with no mitigation techniques applied, 38,660 hazard events were recorded. The implementation of the blacklisting technique marked a significant decrease in these events to 33,970, indicating a 12% reduction. This finding not only underscores the immediate efficacy of the blacklisting approach but also highlights its potential in reducing

hazard events.

Applying the geofencing technique further contributed to hazard mitigation, decreasing to 32,203 hazard events. This represents a 17% reduction compared to the original state, attesting to the effectiveness of this technique, particularly in its capacity to cluster and regulate messages in clustered areas.

The preliminary experiment shows some superiority of prevention against mitigation. This apparent superiority of the preventative approach can be ascribed to the temporal efficiency of machine learning-based methods utilized in endgame prevention, as compared to the more time-intensive nature of mitigation techniques. Specifically, the latency inherent in mitigation processes is highlighted by the duration required for the reputation function to accumulate sufficient data for the blacklisting of malicious vehicles and the time needed to establish geofenced areas. Such delays in the mitigation response could extend the overall duration required to neutralize threats, thereby underscoring the relative advantage of the preventative strategy in this experimental framework.

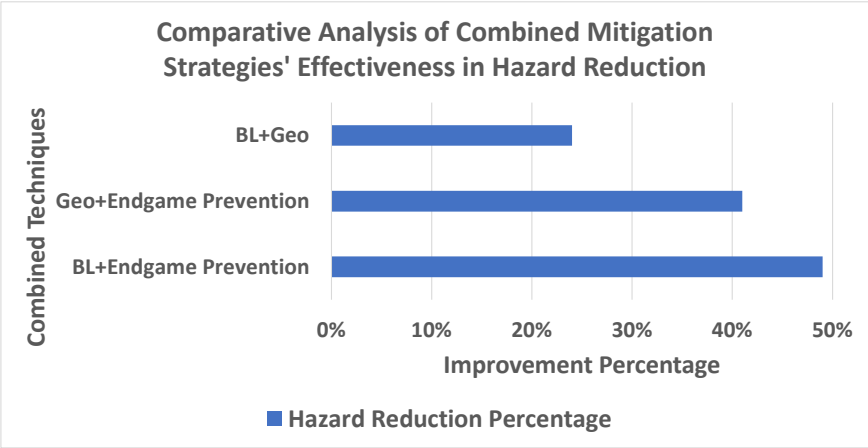


Figure 5.10: Enhancement ratios in hazard mitigation through combined mitigation strategies in VANETs

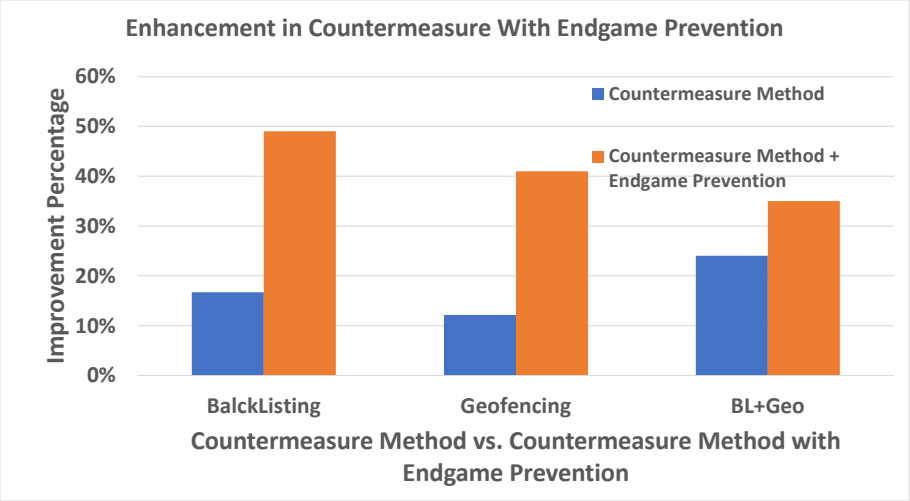


Figure 5.11: Synergistic efficacy of endgame prevention in combination with different mitigation techniques for hazard reduction

5.5.2 Combinatorial Approaches

In this experiment, we have two distinct types of experiments. The first set of experiments investigates the synergistic effects of combining various attack endgame mitigation techniques. The second set focuses on integrating a singular attack endgame mitigation technique with attack prevention measures. Both experimental approaches aim to discern the efficacy of these techniques in isolation and in concert, thereby providing insights into the optimal configurations for VANET security enhancement.

Evaluations of Combined Mitigation Techniques

The experiment on Combinatorial Approaches investigated the synergistic impacts arising from integrating of diverse mitigation techniques, shedding light on the enhanced efficacy derived from their combined application in addressing hazard events.

Figure 5.10 compares different methods used in VANETs to reduce hazards. It shows the percentage increase in hazard reduction achieved when various mitigation

strategies are used together.

The initial findings indicate that combining blacklisting (referred to as BL in the figure) with endgame prevention culminates in a substantial reduction of hazard events, with a recorded decrease of 49%. Meanwhile, using geofencing (noted as Geo in the figure) alongside endgame prevention achieves a 41% reduction in hazard events. This result highlights the effective interplay between the proactive cessation of potentially hazardous messages via machine learning predictions constituting endgame prevention and the active dismissal of communications from confirmed malevolent sources using blacklisting.

Conversely, when endgame prevention is excluded from the strategy and blacklisting is merged with geofencing, the outcome manifests as a 24% decrease in hazard events. This is an improvement, though it is less substantial than other combined techniques. This difference highlights that using the endgame prevention technique alongside other strategies significantly boosts the effectiveness of the mitigation outcome.

Ultimately, these combinatorial insights highlight the value of deploying a multi-angled approach to mitigation. Individual techniques carry specific merits, yet their collective deployment reveals augmented dimensions of defense against hazards, emphasizing the strategic advantage of a multi-faceted mitigation framework.

Augmentation with Endgame Prevention

The findings derived from this experiment underscore the prominence of endgame prevention, prompting a dedicated experimental analysis to assess the augmentative

potential of various countermeasure methods when integrated with endgame prevention. This evaluation provides insights into the extent to which the predictive faculties of endgame prevention can bolster the efficacy of other mitigation techniques. Such an analysis is not only instrumental in enhancing the overall security framework but also pivotal in understanding the synergistic interactions between different defensive strategies.

The analysis, as illustrated in Figure 5.11, reveals a consistent pattern: endgame prevention invariably boosts the effectiveness of other countermeasures. Notably, its combination with blacklisting elevates the hazard reduction rate from 17% (with blacklisting alone) to an impressive 49%. This substantial increase highlights the synergistic relationship between the two methods.

Similarly, combining geofencing with endgame prevention also demonstrates a remarkable improvement. geofencing alone accounts for a 12% reduction in hazards, but this figure ascends to 41% as indicated in Figure 5.11, when coupled with endgame prevention. The strategic approach of geofencing, involving the clustering and regulation of messages in designated suspicious zones, is amplified by the insight provided by endgame prevention.

When blacklisting and geofencing security techniques are used together, they can reduce hazards by 24%. When endgame prevention is added to this combination, the hazard reduction increases to 35%. Although this improvement is good, it is not as significant as the improvement seen in other combinations, which suggests that there may be some overlap when all three techniques are used together.

In conclusion, these findings underline the transformative impact of endgame prevention. While it is effective on its own, its value lies in enhancing the performance of

other countermeasures, strengthening the overall framework for mitigating potential hazards.

5.5.3 Sensitivity and Threshold Analysis

To ensure that the model is effective and adaptable, evaluating its performance across various settings is crucial. This section focuses on examining how well the system adjusts to changes in threshold values, especially in the areas of clustering and blacklisting. By altering these threshold levels, we aim to test the model's stability and determine the best settings for its key parameters. This approach allows us to understand how the model responds under various conditions and guides us in fine-tuning it for optimal performance.

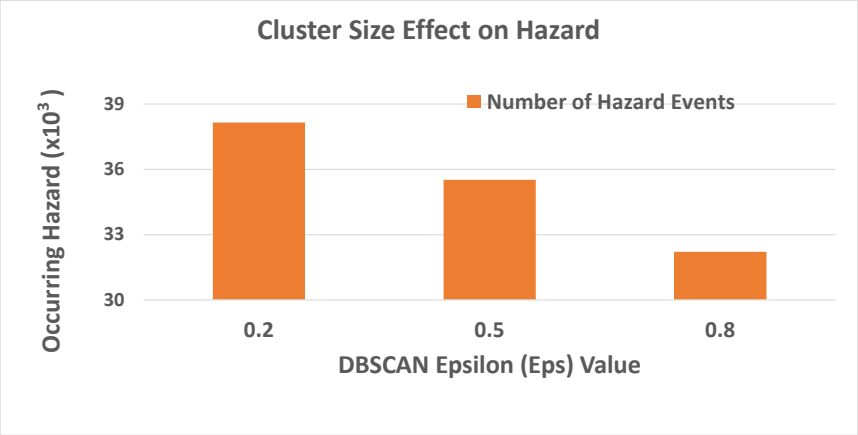


Figure 5.12: Correlation between Epsilon and Min Core values on cluster formation and area coverage in geofencing

Impact of Geofencing Parameters: Min Core and Epsilon Analysis

This experiment aimed to optimize the geofencing technique, which uses the DBSCAN clustering algorithm, by examining its main parameters: Min Core and Epsilon (Eps

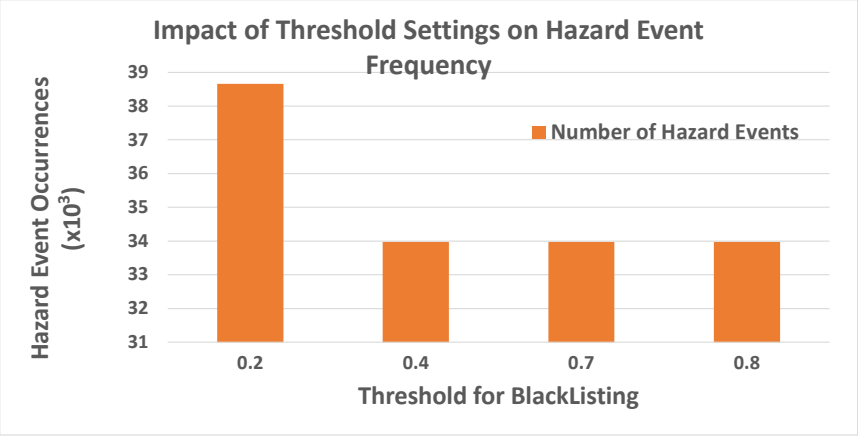


Figure 5.13: Effect of threshold adjustment on hazard event frequency in blacklisting strategy

Eps Value	Min Core Points	Number of Geofenced Areas	Number of Affected Vehicles	Total Duration Time for Geofenced Areas	Hazard
0.2	20	9	381	44	38157
0.5	12	21	706	709	35516
0.8	7	24	743	752	32203

Table 5.1: Impact of parameter adjustments on geofencing efficacy and duration on hazard events

Value). Min Core is the minimum number of points needed for a dense cluster, while Epsilon is the maximum distance for considering points as neighbors.

Figure 5.12 shows the inverse correlation between the number of hazardous events and the increment in the Epsilon value, which correlates with an expanded geofenced area. The number of hazard events consistently decreases with these parameter adjustments. Initially, with an Epsilon of 0.2 and a Min Core of 20, the number of hazardous events is 38,157; this decreases to 32,203 as the Epsilon is raised to 0.8 and the Min Core is lowered to 7.

Table 5.1 shows an escalation of the Epsilon value from 0.2 to 0.8, resulting in a proliferation of clusters, signifying a more granular division of the area. Conversely, a reduction in the Min Core Points from 20 to 7 indicates a decreased threshold for cluster formation, albeit over a more extensive area due to the increased Epsilon. The data from the table further reveal that an increase in Epsilon and a decrease in Min Core Points correspond with an augmented impact of geofencing on the vehicular network, evidenced by an increase in affected vehicles from 381 to 743 and a prolongation of geofencing activity from 444 to 752 seconds. This demonstrates that alterations in these parameters not only extend the coverage area but also the duration of geofencing, which might not be optimal under certain scenarios, necessitates careful consideration. Hence, the selection of an optimal combination of Epsilon and Min Core Points is relegated to the discretion of the administrator, who must weigh the acceptability of the increased geofencing duration against the efficacy in hazard reduction.

Adaptive Thresholding in Dynamic Blacklisting: Hazard Event Analysis

In the process of blacklisting, setting the right threshold is key to determining when a sender should be marked as malicious. We investigate how different threshold settings impact the system's success in reducing hazard events.

Figure 5.13 shows that starting with a threshold of 0.2, which is quite forgiving, leads to 38,660 hazard events. This high number indicates that many harmful messages are still getting through because the threshold is not strict enough. However, when we increase the threshold to 0.4, there is a noticeable decrease in hazard events to 33,970, suggesting that a higher threshold is more effective at reducing hazard

events.

Raising the threshold to 0.7 and 0.8 does not reduce the number of hazard events; it remains at 33,970. This finding suggests that once the threshold reaches a certain level (here, 0.4), raising the threshold does not increase the number of hazards.

This experiment highlights the importance of choosing the blacklisting threshold carefully. While a very low threshold can lead to too many risks, setting it extremely high does not necessarily bring extra benefits. The data indicates that a threshold around 0.4 is optimal for effectively minimizing hazard events.

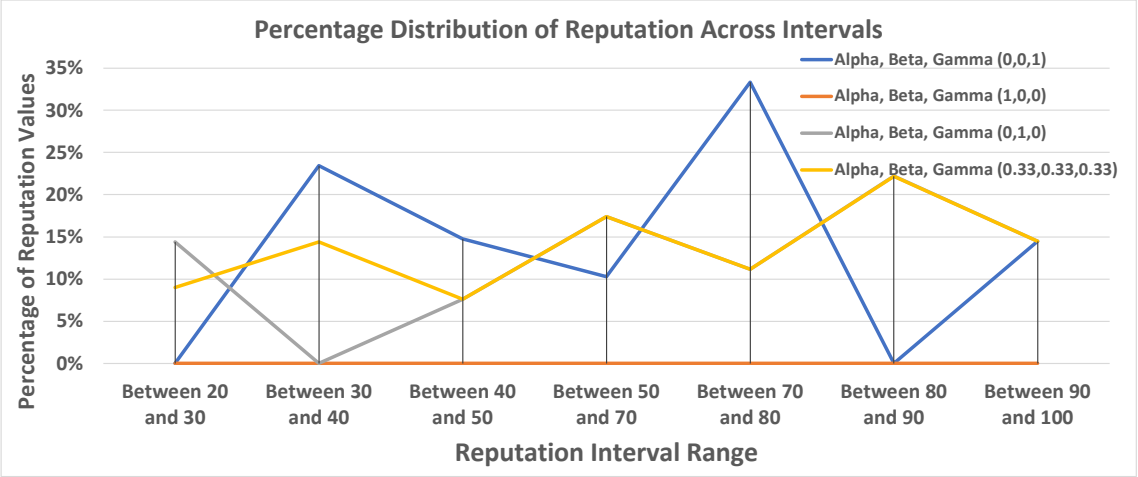


Figure 5.14: Analysis of reputation value distribution across different parameters in RSMA blacklisting framework

5.5.4 Reputation Function in RSMA Blacklisting Technique

We conduct a detailed empirical analysis of an enhanced blacklisting technique by modifying the reputation function used in RSMA. The reputation function is articulated as follows:

$$\begin{aligned} RS_{\text{new}} = \beta \times \text{ObjectiveEvaluation} \\ + \gamma \times \text{EffectDegree} \\ + \alpha \times \text{HistoricalReputation} \end{aligned} \tag{5.3}$$

The primary objective was to evaluate the individual and collective impacts of three key components: historical reputation, objective evaluation, and effect degree on the calculated reputation score. This evaluation involved varying the weights of these components, examining scenarios where one factor was emphasized exclusively while others were minimized and a synergistic balanced scenario where all factors were weighted equally. Figure 5.14 presents the distribution of reputation values under different parameter settings:

1. **Impact of Objective Evaluation Alone**: The parameter setting $(0, 1, 0)$, accentuating objective evaluation, consistently yields lower reputation values, underscoring its relatively minimal impact when functioning in isolation.

2. **Impact of Isolated Effect Degree**: The parameter setting $(0, 0, 1)$ prioritizing effect degree, shows variable performance, indicating that when isolated, the effect degree can lead to fluctuating reputation scores, showcasing its dynamic nature.

3. **Impact of Historical Reputation Alone**: With parameters set to $(1, 0, 0)$, focusing exclusively on historical reputation, the results indicate a consistent distribution of reputation values. This uniformity stems from the factor's reflection of accumulated historical data, maintaining its original value even when other components are nullified, thus highlighting its stabilizing effect.

4. **Integration of All Parameters**: The balanced parameter setting $(0.33, 0.33, 0.33)$, integrating all three components equally, results in a well-distributed range of reputation values. This suggests a synergistic effect achieved through a holistic consideration of all factors.

These insights highlight the distinct roles played by historical reputation, objective evaluation, and effect degree in determining reputation scores within the RSMA blacklisting framework.

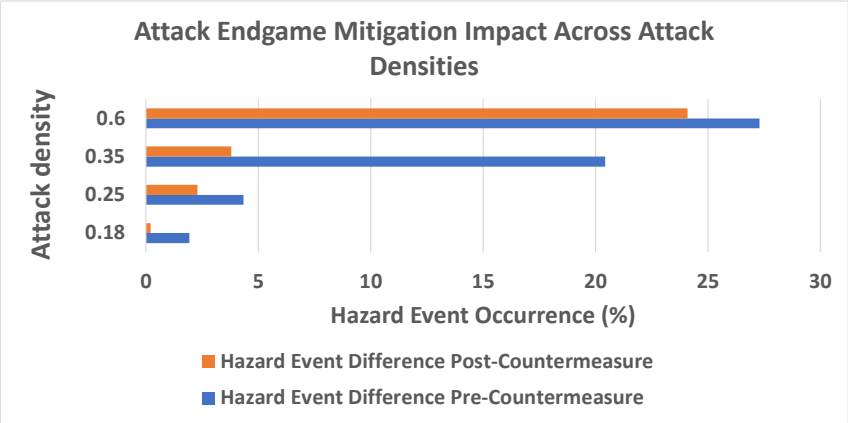


Figure 5.15: Comparative impact of countermeasures on attack endgame across varying attack densities

5.5.5 Mitigation Efficacy Across Diverse Attack Densities

This experiment was designed to examine the adaptability of the proposed mitigation strategies across varying levels of attack density. The primary objective was to understand the influence of fluctuating attack prevalence on the system's susceptibility and to evaluate the effectiveness of the countermeasures in reducing the occurrence of hazardous events.

Figure 5.15 presents a comparative analysis of the impact of countermeasures on the attack endgame within different attack density environments, quantified by the variation in hazard event occurrences. The graph delineates the scenarios before and after implementing countermeasures, indicated by light blue and dark blue, respectively. In the pre-countermeasure state, there is a gradual escalation in attack density corresponding to the increase in hazard event occurrences, peaking at approximately 27.28% for the highest level of hazard events noted. Conversely, following the deployment of countermeasures, there is a conspicuous decrease in attack density across all evaluated categories. This is exemplified by the data point at a 5% hazard event occurrence, where the attack density diminishes from 4.34 to 2.28 after countermeasure implementation, signifying a substantial mitigation impact.

In summary, these insights emphasize the dynamic adaptability of the mitigation strategies, demonstrating their varying degrees of efficacy across different attack densities. Particularly at an attack density of 0.25, they show the necessity of fine-tuning the countermeasures to enhance their responsiveness and effectiveness in mitigating hazards during phases of transitional attack prevalence.

5.5.6 Analyzing the Impact of Mitigation on Message Exchange and Speed Recovery

This section examines how the proposed model affects key operational aspects: the exchange of messages and the recovery of vehicle speeds. These experiments highlight the real-world value and benefits of our mitigation strategies.

Table 5.2 shows the findings on the interplay between mitigation techniques on message count variation, attack message ratio, hazard event ratio, and their collective

influence on the speed recovery ratio under different mitigation strategies.

Table 5.2: Effectiveness of mitigation techniques against fake accident attack on speed recovery

Mitigation Applied	**Original**	**Blacklisting**	**Geofencing**	**Endgame Prevention**	**ALL**
Message Count Variation (%)	0.0	0.0243	-0.0138	-0.0143	-0.0263
Attack Message Ratio (%)	0.0042	0.0045	0.0043	0.0043	0.0048
Hazard Event Ratio (%)	0.0434	0.0372	0.0366	0.0277	0.0289
Speed Recovery Ratio (%)	1.0	1.2	1.4	2.2	3.0

1. Message Count Variation: The implementation of geofencing, endgame prevention, and the combined application of all strategies result in a decrease in message count. This reduction, particularly notable in the combined approach (ALL) category with -0.0263%, suggests that a comprehensive mitigation strategy might lead to more efficient message handling, thereby positively impacting speed recovery. Interestingly, the variation in message count shows a slight increase with the application of blacklisting, indicating a marginal elevation in message traffic which will be considered in our future work.

2. Attack Message Ratio: Across all mitigation methods, there is a slight increase in the attack message ratio, with the combined approach showing the highest increase to 0.0048%. This increment might be indicative of the system's heightened sensitivity in detecting attack messages, thereby enhancing the overall robustness of the network and contributing to improved speed recovery.

3. Hazard Event Ratio: A notable decrease in the hazard event ratio is observed with each mitigation strategy, particularly with endgame prevention and the combined application of all methods. The decrease to 0.0277% and 0.0289%, respectively, underscores the effectiveness of these strategies in mitigating hazard events,

which is a crucial factor leading to an enhanced speed recovery ratio.

4. Speed Recovery Ratio: The speed recovery ratio significantly improves under each mitigation strategy, with the most pronounced improvement observed when all strategies are employed simultaneously (a 200% increase to 3.0%).

In summary, the results suggest that while individual mitigation strategies like blacklisting, geofencing, and endgame prevention each contribute to improving the speed recovery ratio, their integrated application amplifies this effect. The combined approach enhances the system's efficiency in handling messages and significantly reduces hazard events, thereby enabling a more robust and rapid recovery of vehicle speeds post-attack.

5.6 Summary

The work in this chapter substantiates the efficacy of a multi-faceted mitigation strategy against cyber-attacks on VANETs. Despite considerable research on various aspects of VANET security, the profound impact of road hazard-inducing attacks, which pose an immediate threat to physical safety, has remained a less researched area, prompting the focused development of solutions to mitigate these critical risks in vehicular networks. Leveraging the VeReMiAP dataset and harnessing recent progress in predicting the effects of attacks, this work puts forward a dual-faceted strategy designed to counteract the culmination of attack scenarios effectively. This dual strategy commences with preventing attack endgames, which is achieved through deploying sophisticated RNN models. These models play a critical role in forecasting the potential consequences of FRA generated by fraudulent CAMs, thus enabling proactive interventions at an early stage. The second phase of the strategy is dedicated to

implementing countermeasures for attack endgames. This involves the integration of geofencing, which utilizes the DBSCAN algorithm for strategic message regulation. In synergy with this, dynamic blacklist management, anchored in a RSMA framework, is adeptly employed. This system is instrumental in identifying and excluding malevolent vehicles from the network, thereby fortifying the security framework of VANETs against potential threats. The amalgamation of endgame prevention, blacklisting, and geofencing techniques has yielded a substantial elevation in the speed recovery ratio, with a remarkable 200% increase when these strategies are deployed together.

Chapter 6

Attack Rate-Based Prediction for Trust Origin

In Chapters 3 and 5, we demonstrated how proactive security strategies effectively address trust subjects and trust services. However, these strategies sometimes struggle with new types of attacks. This chapter focuses on trust origin to handle unexpected attacks. Here, we complete setting up the components of the proactive TMS and improve the system's strength against new and unforeseen security challenges.

6.1 Introduction

Securing message exchanges within VANETs presents a formidable challenge attributed to the network's distinctive characteristics [123]. Nonetheless, attacks on VANETs can be broadly categorized like other networks, distinguishing between insider [124] and outsider [114], active and passive, or rational and malicious attacks [125, 26]. Each category encompasses specific attack types, including mobility-based, network-based, or identity and traffic manipulation-based attacks.

An attack rate, defined as the frequency of attacks within a given timeframe [126], is a crucial metric for predicting potential attacks or identifying ongoing network threats. This metric can be analyzed from various perspectives, including that of the

attacker or the victim, or through a more granular lens, such as the attack interarrival time, which sheds light on the intervals between consecutive attacks [127].

A honeypot is a deliberately designed computer system that acts as a decoy to lure cyberattacks. It simulates a legitimate target for attackers, allowing security professionals to monitor hacking attempts, gather insights into cybercriminal behavior, and potentially divert attention away from more critical systems. This strategy helps in studying attackers' methods and improving defenses without risking important assets. In addition to these benefits, honeypots have emerged as pivotal tools for gathering data on attack rates. While their primary function is not to directly defend systems, their true value lies in being probed, attacked, or compromised, as they offer critical data for understanding attack patterns [128].

The main contributions of this chapter are to discuss the following:

- Enhance VANET security through strategic deployment of honeypots, a less researched area, by developing a system model that actively uses honeypots for calculating attack rates and identifying threats.

- Facilitate an attack prediction mechanism, utilizing attack rate extrapolated from data on prepared-for attacks to anticipate unprepared-for attacks.

- Utilize dynamic honeypots suited to VANET's changing environment that act passively to avoid disseminating false messages, focusing on interaction patterns without revealing vehicle identities.

- Validate the approach's effectiveness with public datasets and various machine learning models, showcasing the system's ability to predict unexpected attacks

without an increase in false positives and demonstrate the optimal use of honeypots for maximum efficiency.

The rest of the chapter is organized as follows. Section 6.2 reviews existing studies on honeypots and methods for calculating attack rates in VANETs. Section 6.3 outlines the problem we aim to solve and introduces key concepts essential to the approach. Section 6.4 thoroughly describes the proposed solution and the proposed algorithm. Section 6.5 details the results of the simulations, providing a detailed assessment and critique of the performance of the proposed method. We conclude the chapter in Section 6.6 by summarizing the main findings.

6.2 Related Work

In this section, we review existing studies on TMS applied to VANETs. We then examine the methods for calculating attack rate and its significance in security. Additionally, we investigate the use of honeypots as a method for determining attack rates and identifying previously unknown attacks.

Hussain et al. [46] examined trust establishment and management strategies in VANETs. Through an analytical review of mechanisms proposed between 2014 and 2019, the work reveals a reliance on variations of existing cryptographic and non-cryptographic methods rather than the introduction of new models despite the evolving complexities of VANETs. Significantly, this work underscores the necessity of entity and data trust in ensuring robust vehicular communication, highlighting the potential of integrating machine learning techniques for trust management. It concludes by identifying a gap in the current literature—the absence of new trust management models.

For cyber attack rates, Fang et al. [33] showed the advancement of a deep learning framework, the bi-directional recurrent neural networks with long short-term memory (BRNN-LSTM), tailored for cyber attack rate prediction. The empirical validation of the framework, employing data from a low-interaction honeypot across five distinct time intervals, demonstrates a significant outperformance over traditional statistical models in terms of prediction accuracy. This leap in predictive performance, facilitated by the BRNN-LSTM's capacity to integrate multiple LSTM layers, addresses the intricate and long-range dependencies in cyber attack data. The work is pivotal, not merely for its methodological innovation but for its practical implications. It offers a more accurate, proactive approach to cyber defense and underscores the critical role of predictive analytics in fortifying network security infrastructures.

Honeypots were also used as a proactive security approach. Patel et al. [129] proposed a proactive IDS leveraging honeypots to combat selfish behaviors in nodes conserving their resources for personal gain. The Honeypot Selfishness Detection (HPSD) method employs a baiting technique to identify such nodes. When a vehicle discards a packet, it is flagged as potentially selfish, and the honeypot's baiting strategy is applied. This method contrasts with the standard AODV routing protocol, wherein a node broadcasts a Route Request (RREQ) packet if no direct path is found. A cooperative node would re-broadcast this to its neighbors. However, if the RREQ is not returned from a neighbor, that neighbor is suspected of being selfish, and a bait RREQ is sent to it. If the node drops this bait, it is confirmed as selfish and excluded from routing paths. This approach has been shown to enhance packet delivery ratios and reduce end-to-end delays, demonstrating the efficacy of honeypots in identifying malicious nodes, with potential for further refinement and broader

application. However, it introduces additional network traffic and overhead through bait messages and is currently limited to detecting only energy-based selfish nodes.

To bridge existing research gaps, particularly concerning system performance and detecting new (zero-day) and known attacks with minimal system burden, Sharma et al. [130] introduced a new IDS framework enhanced by honeypot technology, Honeypot optimized IDS (HPIDS). This framework combined traditional signature-based IDS for known threats with honeypot nodes that attract and log malicious activities, generating signatures for new threats. This innovative approach strengthened the system's ability to detect various attacks. Despite progress in VANET security, this research highlighted significant issues, such as where to deploy IDS, choosing the right IDS architecture, and balancing detection accuracy with resource use. It pointed out that current research mainly covers specific attack types and lacks clarity on the best detection and placement strategies for VANETs. Furthermore, there is a notable diversity in how research validations are conducted and little exploration into how IDS could be effectively implemented in VANET Cloud environments for detecting malicious activities.

Zobal et al. [131] categorized honeypots into various types, offering examples of historically influential ones and those currently under development. They acknowledged that while honeypots are not a panacea for all cybersecurity challenges, their unique capabilities afford valuable insights into adversary behaviors, facilitating rapid responses to emerging threats. The paper emphasized the necessity for honeypots to become more user-friendly in deployment and maintenance to remain a vital asset within the cybersecurity toolkit. Additionally, it highlighted the importance of honeypots being able to evade detection by adversaries and the need for sophisticated

data analysis tools that offer clear visualization and intelligent clustering of attack data to cope with the complexities of the modern data-rich cybersecurity landscape. This research suggests that honeypots, capable of simulating vulnerabilities without posing an actual risk, are a critical component in the cybersecurity defense arsenal cyber defense arsenal. However, they must evolve quickly to meet the demands of a rapidly changing cyber threat environment.

6.3 Preliminaries

Detecting unexpected or zero-day attacks poses a significant challenge in implementing forward-thinking security strategies. Using honeypots has been identified as a particularly effective solution to this problem [130]. This section outlines the various challenges in predicting such attacks and describes the key ideas behind the proposed solution.

6.3.1 Trust Origin

Trust origin does not have a clear definition, but it can be seen as the degree of confidence that the other party will act socially responsibly [132]. There are two critical roles in any situation involving trust: the trust giver and the receiver. The trust giver is the one who decides to place trust, and the trust receiver is the one who is trusted. This relationship is crucial for making decisions and carrying out actions, especially when the outcomes are uncertain and carry significant weight [133].

In the research of TMS, work often centers on the trust subject [134] referring to the entity or the content and the trust service [135] related to aspects of security or informatics. However, a crucial aspect frequently overlooked is the trust origin,

which assesses the reliability of the connections through which trust is established. The definition and emphasis on trust origin are inconsistent across all TMS, as its interpretation largely hinges on the system's specific architecture and the foundational principles of the trust model it adopts.

Trust origin is categorized into three distinct types: Direct, involving information sourced directly from the trust subject, such as self-reported data; Indirect, which relies on trust judgments based on the recommendations or endorsements from entities considered to be trustworthy; and aggregate, where data from various sources are merged to provide a complete understanding.

The essence of examining trust origin lies in verifying the network's security status, focusing on direct and indirect methods to establish trust. This process is not limited to addressing specific attacks but extends to determining whether the network's entities are reliable sources for trust calculation. This nuanced examination highlights the significance of recognizing and evaluating the foundations and credibility of trust-establishing connections within a network's framework.

6.3.2 Attack Rate

To understand trust origin effectively, a method that goes beyond pinpointing specific attacks and offers a broader view of the network's security is essential. The Attack Rate is particularly suitable for this purpose, as it provides crucial insights into the overall state of the network and gauges the reliability of vehicles for future interactions.

The Attack Rate acts as an early warning system, where an elevated rate may indicate potential risks, prompting further analysis of a vehicle's actions. Additionally,

it allows for the real-time adjustment of trust levels assigned to vehicles, updating these scores to reflect their current actions. This chapter primarily examines how the Attack Rate can be an initial step in evaluating trust.

6.3.3 Honeypots

Gathering data on attack rate can be efficiently achieved using honeypots in the network [136]. Honeypots are digital traps designed to attract attackers, diverting their attention from real targets while collecting important information about their strategies and methods [137]. Key features of honeypots include acting as decoy systems that imitate real systems, often using fake data or services to attract attackers. They monitor interactions by logging and observing all activity for later analysis. Honeypots come in different types, from low-interaction versions with basic functionality to high-interaction ones that closely mimic real systems. Each type carries different levels of risk and engagement with attackers.

In the presented model, honeypots act as virtual vehicles within the network, created and managed by a trusted third party. These digital vehicles, known as honeypots, participate in the network by exchanging BSMs like real vehicles do. The honeypots are set to operate passively, meaning they will not spread incorrect information, thus ensuring the safety of actual vehicles. Their purpose is to attract and receive false messages without questioning the sender's authenticity while gathering and verifying messages from nearby vehicles. This strategy is particularly valuable when the TTA cannot oversee all message exchanges in the network or aims to focus on specific areas. Furthermore, processing fewer messages allows these honeypots to detect and predict issues quickly, improving the system's security response.

Incorporating honeypots within VANET environments presents specific challenges:

1. **Mobility and Disconnection**: Constant movement and frequent disconnections of vehicles in VANETs complicate stable engagement with honeypots.

2. **Limited Interaction**: VANETs rely on concise, safety-related messages, which reduces opportunities for detailed honeypot interactions.

3. **Deception in Safety-Critical Systems**: The use of deception raises ethical concerns in systems where safety communication is crucial.

4. **False Positives and Misinterpretation**: Distinguishing between malicious actions and benign errors or unexpected behaviors in VANETs adds complexity.

These challenges highlight the need for careful consideration and innovative solutions when deploying honeypots in VANET environments.

6.3.4 Distinguishing Attack Types for Improved Attack Rate Analysis

To utilize honeypot-collected data on attack rate effectively, we distinguish between two kinds of attacks: those we are ready for (prepared-for attacks) and those we are not (unprepared-for attacks). Prepared-for attacks are the ones that the honeypot's IDS or Intrusion Prediction System (IPS) is designed to guard against, identified by known attack patterns. Unprepared-for attacks, however, are potential threats that the honeypots lack prior knowledge of. This leads us to investigate whether there is a connection between different types of attacks in VANETs and how this knowledge could help us.

For this exploration, we turned to the VeReMi extension dataset [71], which provides a detailed set of attacks generated in VANET simulations. We modified its

classification to suit the proposed work better, dividing the attacks into basic integrity disruptions and advanced persistent attacks. Basic integrity disruptions are related to position and speed (such as constant or random changes), while advanced persistent attacks include more complex strategies such as delayed messages, DoS, data replay, and others.

Our revised classification aims to align with the specific vulnerabilities of VANETs. Since these networks are more susceptible to basic integrity disruptions, choosing a prepared-for attack from this category is advised due to the straightforwardness of implementing detection measures. Focusing on advanced persistent attacks is recommended for unprepared-for attacks because of their complexity and significant threat. This approach not only simplifies understanding of the various attacks in VANETs but also guides the enhancement of honeypot-based security strategies.

6.4 Proposed Scheme

The proposed system is designed to fortify network security by leveraging advanced data analytics and machine learning technologies. This section outlines the core components of the system and the methodologies employed to predict prepared-for and unprepared-for attacks.

6.4.1 System Overview

Figure 6.1 provides a detailed exposition of the architecture and operational intricacies of the proposed system, which is designed to function across three distinct tiers: the vehicular, edge, and cloud layers, and operates in two specific modes: general and honeypot. In the general mode, data is aggregated from the entirety of the vehicular

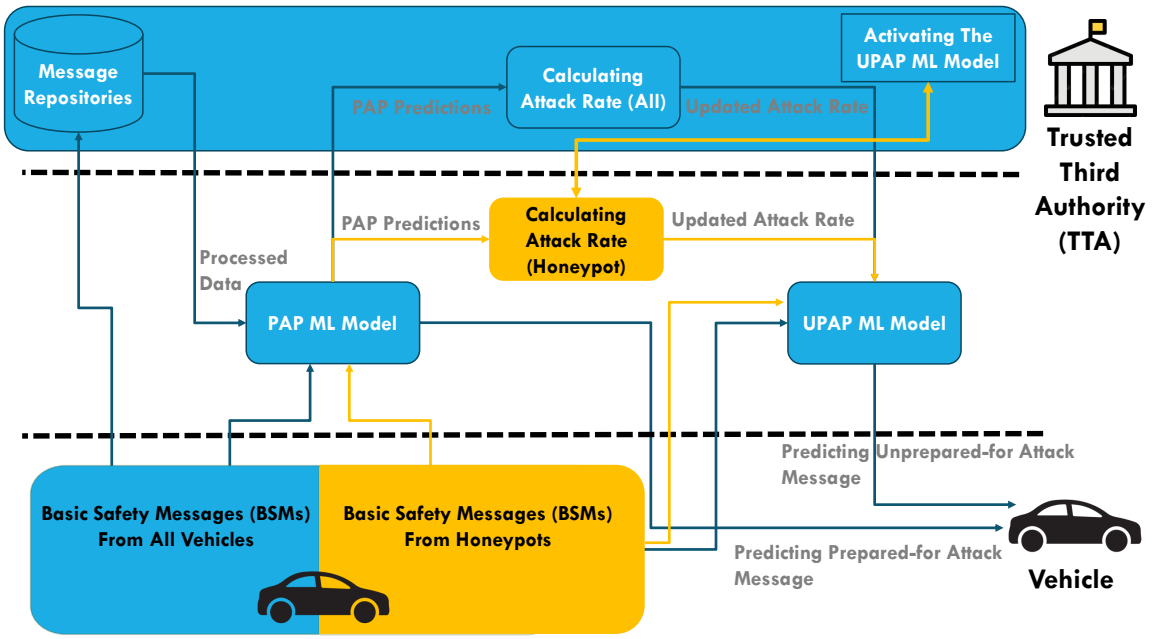


Figure 6.1: Architecture of the proposed VANET security system: showing the three-tier (Vehicular, Edge, Cloud) structure and dual modes (general and honeypot)

network, whereas the honeypot mode exclusively processes data from strategically designated honeypots. At the vehicular tier, BSMs are collected. These BSMs are directed towards two principal destinations. Initially, the BSMs are transmitted to a message repository located within the cloud infrastructure under the surveillance of a TTA. This repository serves as a collection point for a predefined duration, after which the accumulated BSMs are utilized to train the Prepared-for Attack Predictor (PAP) ML model. This strategic accumulation of messages aims to furnish the PAP model with a robust dataset for training purposes. Subsequently, live BSMs exchanged amongst vehicles are relayed to the PAP to detect anticipated attacks.

The methodology for calculating the attack rate is contingent upon the operational mode of the system. In the general mode, the TTA calculates the attack rate

employing prediction data from the PAP ML model at the cloud tier. Conversely, attack rate calculations are executed at the edge tier in the honeypot mode, leveraging predictions from the PAP but solely based on honeypot data. The communication link between the TTA and the honeypot-based attack rate calculation component plays a crucial role in transmitting the data collection status and in orchestrating the activation of the Unprepared-for Attack Predictor (UPAP) model. Upon accumulating a sufficient dataset, the TTA activates the UPAP model for both modes, enabling the integration of attack rate-based trust scores into the UPAP model. The UPAP model then begins analyzing exchanged BSMs to identify potential unprepared-for or ongoing attacks. In conclusion, the system ensures that detections from the PAP and predictions from the UPAP model are efficiently disseminated back to the vehicular network, thereby enhancing the security posture of the entire ecosystem.

6.4.2 System Modules

Data Collection and Preprocessing Module

The system core is built on a solid data collection and preprocessing module. This part of the system collects essential information such as network traffic data, attack patterns, and honeypot records. It handles essential tasks including cleaning the data, organizing it, and identifying key features. After preparing the data, it is divided into separate groups for training and testing. This step is crucial for training the models and checking the accuracy of the prediction models.

Prepared-for Attack Predictor (PAP) Machine Learning Model

The heart of the system is the PAP model. This machine learning model constantly works to recognize and address known attacks. This model is crafted to spot and counteract familiar threats by examining patterns and identifying marks in network traffic. It serves as an initial layer of protection against well-documented cyber threats.

Attack Rate Calculation Module

A novel system component is the attack rate calculation module, which processes historical attack data to calculate the frequency and intensity of ongoing known attacks. This calculation can be conducted either by a centralized third authority or through a distributed network of honeypots, which also collect data on interactions with potential attackers for further analysis. The insights gained from this module will be used later to enhance the prediction accuracy of the next model by providing an understanding of attack patterns and trends.

Unprepared-for Attack Predictor (UPAP) Machine Learning Model

The system features the UPAP model, which tackles the issue of unknown or unexpected attacks. This machine learning model is specially trained on data that includes attack rate information from the attack rate calculation module. The UPAP model aims to foresee attacks without relying on usual indicators by using attack rates to highlight potential weak spots and the chance of unforeseen attacks on the network. This ability to predict is vital for broadening the system's protective measures to include a broader range of attacks, especially those without previously known signatures or patterns.

6.4.3 System Description

Algorithm 6.1 outlines a method for calculating attack rate using honeypot data, an essential step in defending against cyber threats. This algorithm is key in predicting attacks for which we are not specifically prepared. The data used by the algorithm **testData** includes the test portion analyzed by a machine learning model to identify known or prepared-for attacks.

The process begins by selecting the most attacked honeypots, labeled as **selected-Honeypot**, from the dataset. These honeypots are chosen based on how often they are attacked. The number of these important honeypots is indicated by **topCount**, typically set to 50 for this example, to represent a sample (lines 3 and 4).

After selecting these honeypots, the algorithm counts the total messages for each one, determining their communication activity. It then calculates how long each honeypot has been active by measuring the time between their first and last messages. This time is converted to hours to standardize the data for further analysis (lines 7 - 11).

The algorithm calculates different attack rates for each machine learning model (lines16 - 21). It looks at a count-based attack rate **countRate**, showing how often attacks happen in relation to the number of messages. It also figures out a time-based attack rate **timeRate** by comparing predicted attacks to how long the honeypot has been communicating. Furthermore, a combined attack rate **combindRate** is determined, blending message count and time data to give an overall metric.

Towards the end, these calculated rates are added to the **testData** dataset, specifically updating the entries for the **selectedHoneypots** (line 22). This addition gives

Algorithm 6.1 Calculate Attack Rates Using Honeypots

Input:
attacksData - Data about attacks on honeypots and other vehicles
topN - Number of top honeypots to be considered
Output:
testData - Dataset with predictions and attack rates from th top honeypots
Variables:
See within procedures

procedure DETERMINE TOP HONEYPOTS
  Determine *selectedHoneypots* from *attacksData* based on *topN*
  topCount ← 50 ▷ Example: top 10 honeypots
  selectedHoneypots ← *topCount* in *attacksData*
end procedure
procedure DATA PREPROCESSING
  Calculate total messages for each *selectedHoneypots*
  totalMsgs ← *testData* filtered by *selectedHoneypots*
  Determine *firstMsgTime* and *lastMsgTime*
  duration ← *lastMsgTime* - *firstMsgTime*
  durationHours ← *duration* / *3600*
end procedure
procedure CALCULATE ATTACK RATES
  for each Machine Learning model **do**
    predictedAttacks ← Number of attacks detected by the model
    Calculate *countRate* for top honeypots
    $countRate \leftarrow \frac{predictedAttacks}{totalMsgs}$
    Calculate *timeRate* for top honeypots
    $timeRate \leftarrow \frac{predictedAttacks}{durationHours}$
    Calculate *combinedRate* for top honeypots
    $combinedRate \leftarrow \frac{countRate}{durationHours}$
    Update *testData* with calculated rates for *selectedHoneypots*
  end for
end procedure

the dataset more depth by providing the next predictive model with detailed information on the frequency and timing of honeypot attacks.

The algorithm improves the model's ability to predict unknown (unprepared-for) attacks, strengthening the cybersecurity system's overall defense.

Algorithm 6.2 Calculating Reputation Based on Attack Rate

Output:
testData - Dataset with predictions and updated reputation scores.
Variables:
rateType - Represents each type of attack rate within the loop.
minRate,*maxRate* - Minimum and maximum values for a specific attack rate type.
normRate - Normalized attack rate.
histReputation - Historical reputation score inversely proportional to the normalized attack rate.

```
1: procedure DefineAttackRateTypes
2:     attackRates ← [
        countRate,
        timeRate,
        combinedRate]
3: end procedure
4: procedure NormalizeAndCalculateReputation
5:     for each rateType in attackRates do
6:         Normalize rateType
7:         minRate ← testData[rateType].min()
8:         maxRate ← testData[rateType].max()
9:         normRate ← (testData[rateType] − minRate)/(maxRate − minRate)
10:        Calculate histReputation inversely proportional
11:        histReputation ← 1 − normRate
12:        Add histReputation to testData
13:        testData[histReputation + rateType] ← histReputation
14:    end for
15: end procedure
```

Algorithm 6.2 presents a method to assess the reputation of network entities based

on their interaction with attacks, playing a crucial role in enhancing cybersecurity measures. This algorithm is pivotal for distinguishing between benign and potentially malicious activities. It leverages data from the dataset **testData**, which includes outcomes analyzed by a predictive model to identify known attacks.

In the initial procedure, *DefineAttackRateTypes*, the algorithm establishes the types of attack rates to be analyzed: **countRate**, **timeRate**, and **combinedRate** (lines 3). This categorization is crucial for a nuanced analysis that accounts for various dimensions of attack data.

Subsequently, in the *NormalizeAndCalculateReputation* procedure, the algorithm iterates over each **rateType** within **attackRates**, normalizing these rates based on the minimum and maximum values found within **testData** (lines 5-9). This normalization process is essential for uniformly adjusting the attack rate data, facilitating a fair comparison across different attack rates. The core of the algorithm's functionality where the **histReputation** score is computed as inversely proportional to **normRate** (lines 10 - 12). This computation underscores the algorithm's approach to reputation scoring, where a higher normalized attack rate inversely impacts the historical reputation score, reflecting a heightened security risk. The final step involves integrating the calculated **histReputation** scores back into **testData**, enriching the dataset with nuanced reputation insights based on attack rate analysis (line 13).

This approach not only refines the dataset for subsequent analytical processes but also underpins a more resilient cybersecurity framework by providing a nuanced understanding of trust within the network. By calculating reputation based on attack rate, the algorithm aids in preemptively identifying entities that may pose a risk, thereby strengthening the network's defense against known and emerging cyber

threats.

Algorithm 6.3 Calculating Trust

Output:
testData - The dataset with updated trust values.
Variables:
obsTrustWeight - The weight for the observed trust.
histRepTypes - List of historical reputation types.
posCount, *negCount* - Numbers of positive and negative outcomes.
obsTrust - Observed trust component calculated.
overallTrust - Overall trust for each historical reputation type.

1: **procedure** InitializeVariables
2: $obsTrustWeight \leftarrow 0.5$ ▷ Define the weight for observed trust
3: Define historical reputation types:
4: $histRepTypes \leftarrow [countRep, timeRep, combRep]$
5: **end procedure**
6: **procedure** CalculateObservedTrust
7: Calculate *posCount* and *negCount* based on model predictions
8: $posCount \leftarrow testData['ML\_predictions']$
9: $negCount \leftarrow 1 - posCount$
10: Calculate the observed trust component:
11: $obsTrust \leftarrow (posCount + 1)/(posCount + negCount + 2)$
12: **end procedure**
13: **procedure** CalculateOverallTrust
14: **for** each *histRep* in *histRepTypes* **do**
15: Calculate the overall trust for each historical reputation type
16: $overallTrust \leftarrow obsTrustWeight \cdot obsTrust + (1 - obsTrustWeight) \cdot testData[histRep]$
17: Update *testData* with *overallTrust*
18: **end for**
19: **end procedure**

Algorithm 6.3 is focused on measuring trust, which combines observed behavior with past reputation to create a detailed trust profile for each network entity. This step is crucial for developing a deeper understanding of trust within the network, considering recent actions and past behavior related to cybersecurity threats.

Initially, the algorithm sets a key parameter, **obsTrustWeight**, at 0.5 to balance the influence of recent behavior against historical reputation (line 2). This balance ensures that the trust calculation considers immediate actions and long-term patterns fairly.

The algorithm then sorts historical reputation into the categories **countRep**, **timeRep**, and **combRep** (lines 4). Each category captures a different aspect of an entity's exposure to cyber threats, allowing for a well-rounded trust assessment.

Next, it calculates **posCount** and **negCount**, which reflect the occurrences of positive (safe) and negative (risky) behaviors based on the predictive model's outcomes (lines 8-9). These figures form the basis for the observed trust component.

The method's core is determining the observed trust, **obsTrust**, using a formula that balances positive and negative actions (line 11). This calculation includes a correction factor to ensure a fair evaluation of recent actions, reducing the skew from outliers or limited data.

For every type of historical reputation, the algorithm combines the observed trust with the entity's past reputation, weighted by **obsTrustWeight**. This step creates an overall trust score, **overallTrust**, which captures a comprehensive view of an entity's reliability by merging recent and historical behavior (lines 14-16).

By integrating these trust scores back into the dataset, **testData**, the algorithm adds a dynamic layer of trust analysis (line 17). This enhancement equips the dataset for further analysis, especially in predicting and addressing cyber threats that have not been previously encountered, reinforcing the network's defense mechanisms.

6.4.4 Addressed Challenges

Our approach tackles several challenges linked to using honeypots in VANETs, aiming for a secure yet responsible method to improve network safety. Firstly, concerning dynamic topology, VANETs' network structure changes frequently because vehicles are always moving. Our strategy involves using honeypots that adapt to these changes, ensuring they can cover a wide area and interact with many users without being restricted by location changes.

Secondly, regarding ethical considerations, introducing honeypots into VANETs can raise ethical issues, especially if fake signals from decoy vehicles endanger safety or disrupt emergency signals. To avoid this, we use passive honeypots that do not send out misleading messages, ensuring the safety of nearby vehicles. While this choice may limit what the honeypots can do, we view it as an important ethical boundary and a topic worth exploring further in research.

Thirdly, in attributing attacks, it is challenging to identify where an attack comes from in a network as variable as VANETs. Trying to pinpoint the attacker based on honeypot data is complex. Our solution focuses less on finding out who the attacker is and more on the types of interactions that occur. We also rely on data from known attacks to help us predict future threats. This approach is based more on the concept of trust origin, which looks at the nature of the interactions rather than trust subject, and focuses on the identity of the interactors, setting our method apart from others.

6.5 Performance Evaluation

In this section, we thoroughly assess the system through carefully planned experiments to understand how well it predicts unexpected attacks using attack rate data

and examine the effectiveness of honeypots in decentralized networks. **Experiment Group A** tests the system's ability to handle unforeseen attacks. We adopt a three-step method to see how attack rate data affects the system: we look at changes in F1 scores under different attack conditions in Section 6.5.1, compare various performance metrics to measure the system's success in Section 6.5.2, and assess how attack rate data enhances the predictive features' relevance in Section 6.5.3. **Experiment Group B** investigates how honeypots perform, focusing on their use in areas with high data exchange in Section 6.5.4 and their placement efficacy on the attack rate prediction accuracy in Section 6.5.5. These studies are essential for demonstrating the system's theoretical basis and showing its practical use and robustness in improving network security against familiar and new threats.

The experiments utilize F2MD [70] applied to the VeReMi dataset extension attacks, conducted within the LuST [83] scenario as integrated into F2MD. These tests ran for 1500 seconds (25 minutes) involving 366 vehicles, with a beacon exchange rate of 1 Hz and an attack probability set at 0.25. This setup allows us to critically evaluate the system's performance in a realistic traffic simulation, offering valuable insights into its efficiency and adaptability in real-world scenarios.

The experiments use random forest and gradient boosting models because of their strong track record in dealing with complex problems such as identifying cyber threats. Random forest is good at working with varied data and helps avoid overfitting, a common issue where models might not perform well on new data. It also highlights the most essential features, especially when we add information like attack rates. Gradient boosting builds on this by gradually improving its predictions, catching details that random forest might miss. It is particularly good at understanding

how different pieces of information work together, which is crucial for spotting attacks. Using both models, we aim to investigate how including attack rate data can improve attack detection.

6.5.1 F1 Score Comparison Across Attacks

In this experiment, we aim to assess how using attack rate data affects the F1 score, an essential measure of how well the predictive models perform across different types of attacks in a VANET setting. We compare F1 scores achieved with and without attack rate data across a wide range of attacks to understand its impact.

Figure 6.2 compares F1 scores for five specific attacks we were unprepared for, demonstrating how the scores change depending on whether attack rate data was used. We found that F1 scores generally improved for all types of attacks when we included attack rate data in our models. This improvement was evident for attacks closely related to those we had prepared for, like the random speed offset attack, which saw a significant increase in its F1 score. However, for attacks without a clear link to the prepared scenarios, such as data replay and DoS disruptive, the enhancements were less dramatic, showing that the benefit of attack rate data can vary depending on the attack's nature. An interesting case was the random Pos attack [16], which was very similar to the prepared scenarios, showing little difference in F1 scores whether we used attack rate data or not. This highlights the challenge of predicting attacks we are not ready for and the specific ways attack rate data can help improve the models' accuracy.

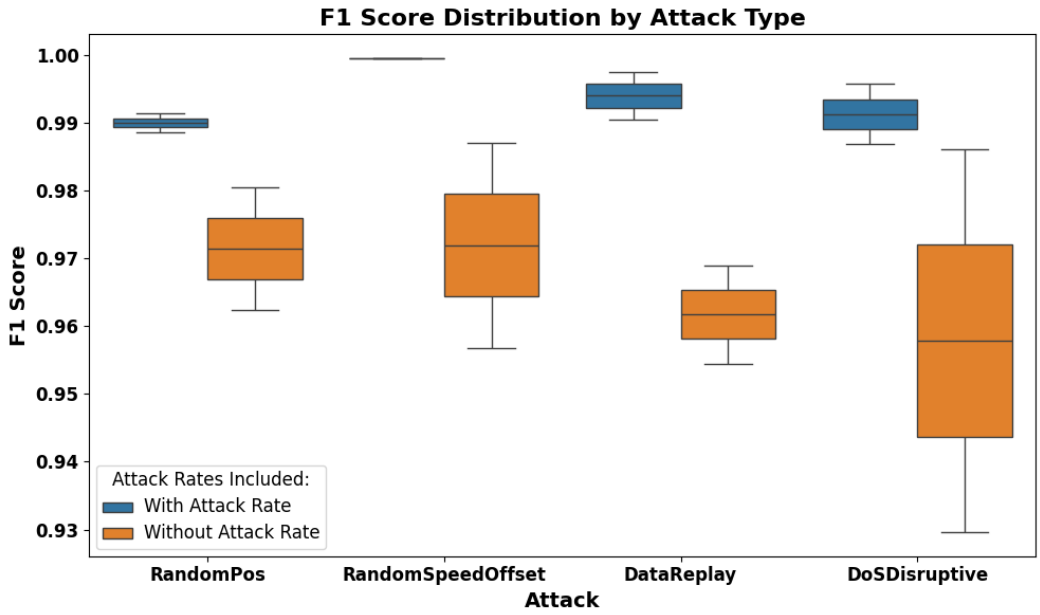


Figure 6.2: Comparison of F1 Scores for five unprepared-for Attacks: illustrating variations with and without the use of attack rate data

6.5.2 Performance Metrics Comparison

In this experiment, the main goal was to see how using attack rate data affects the predictability of unexpected attacks by the models. We assessed this impact using four important measures: accuracy, precision, recall, and F1 score. By comparing model performances with and without attack rate data, we aimed to understand how this data influences the models' ability to predict various attacks. As presented in 6.3, the summarized findings highlight how attack rate data affects these measures.

The analysis focused on models trained with data from the constant position attack and how they performed against different unexpected attacks, including data replay, DoS disruptive, random position, and random speed offset attacks. We demonstrated how leaving out attack rate information affects the models' effectiveness against attacks that closely resemble or differ significantly from the constant

position attack.

The findings reveal a clear difference in model performance for attacks quite different from the constant position attack, such as data replay and DoS disruptive attacks. Without attack rate data, the models were less accurate and less precise in identifying these attacks. This drop in performance is due to the fundamental differences between these attacks and the constant position attack, with the lack of attack rate data widening this gap further.

However, for attacks similar to the constant position attack, like random position and random speed offset attacks, the models still performed relatively well even without attack rate data. This is primarily due to the fact that these attacks exhibit similar patterns, which the models trained on constant position attack data are still able to identify.

Adding attack rate data significantly improved accuracy across all types of attacks, even those initially challenging to detect, like data replay and DoS disruptive attacks. This shows that attack rate data gives the models a more detailed context, helping them distinguish between normal and suspicious activities more effectively. For attacks related to the constant position attack, attack rate data maintained model performance and enhanced it, proving its value in making predictions more precise and reliable.

6.5.3 Feature Effectiveness With/Without Attack Rate

This experiment aims to pinpoint the most critical features that influence the performance of two classifiers, random forest and gradient boosting, especially when considering attack rate data. Our approach involves a detailed comparison to assess

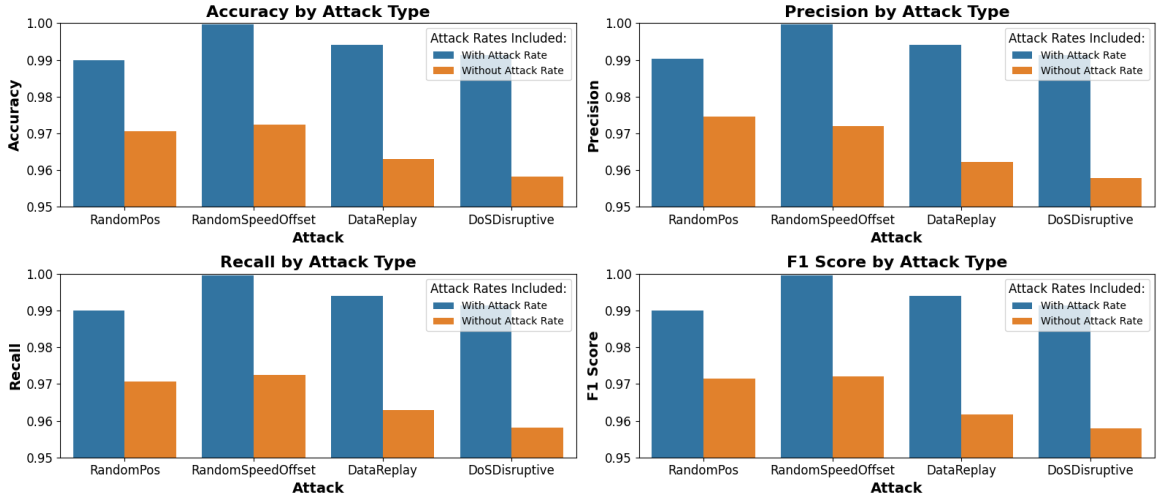


Figure 6.3: Impact of attack rate data on model performance, showcasing changes in accuracy, precision, recall, and F1 score when predicting unexpected attacks with versus without attack rate data

how the importance of top features changes when attack rate information is included or excluded in these models. This method allows us to show the specific role of attack rate data in improving the models' ability to predict unprepared-for attacks effectively. By examining the impact of attack rate data on feature significance within random forest and gradient boosting classifiers, this analysis seeks to reveal the key elements that enhance the classifiers' predictive capabilities. Through this comparative evaluation, we aim to show how incorporating attack rate data can refine the models' focus on specific features, thereby improving their accuracy in identifying potential cyber threats. This investigation offers insights into optimizing feature selection for better model performance and highlights the added value of attack rate data in cyber threat detection strategies.

Figure 6.4 illustrates the significance of specific features, such as changes in position and speed and the vehicle's current location, in predicting unprepared-for attacks

without attack rate. Considering that the machine learning models were not initially built for these attacks. The discussions in Sections 6.5.1 and 6.5.2 reveal that these features perform adequately in scenarios involving attacks similar to those for which the model was trained. However, their effectiveness diminishes when facing attacks of a different nature.

In contrast, Figure 6.5 highlights the model's improved success in predicting these unexpected attacks by incorporating a measure of calculated trust informed by the attack rate data. The results from the earlier experiments, as mentioned in Sections 6.5.1 and 6.5.2, indicate that utilizing features developed from the attack rate data enhances the model's accuracy in detecting attacks similar to the training scenarios and in identifying those that are not. This improvement in how well the model predicts attacks highlights the value of adding flexible, data-driven features that use information from attack rates. By doing this, we give the model a deeper understanding of the context around attacks, which helps it more accurately identify a broader range of cyber threats.

6.5.4 Honeypots with High Exchange Data Rate

This experiment investigates the relationship between the effectiveness of predicting the unprepared-for attacks, as quantified by the F1 score, and the deployment of honeypots characterized by high data exchange rates. Specifically, the experiment aims to elucidate the impact of increasing the number of high-exchange honeypots on the detection capabilities of random forest and gradient boosting. A dual-phase analysis is conducted for each model to discern the influence of incorporating attack rate data into the predictive process versus not incorporating attack rate.

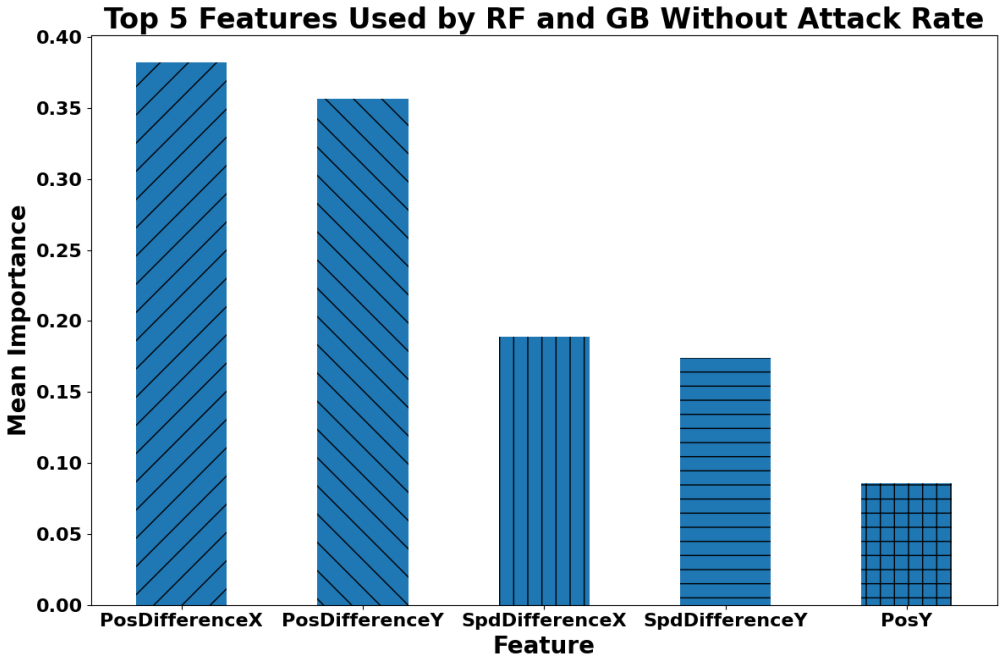


Figure 6.4: Significance of feature variables in gradient boosting model for predicting unprepared-for attacks, highlighting the role of position, speed changes, and vehicle location

By comparing the performance outcomes of the random forest and gradient boosting models under varying honeypot densities and configurations, this experiment sheds light on the optimal use of honeypots in enhancing model sensitivity and specificity. Notably, including attack rate data is pivotal in augmenting the predictive accuracy, revealing significant differences in model performance with and without this contextual information.

Figure 6.6 illustrates the enhancement in the F1 scores for the random forest and gradient boosting models when attack rate data is utilized to predict the unprepared-for attacks. This improvement underscores the significance of augmenting the predictive process with data regarding attack rates derived from a broader array of sources. Conversely, in scenarios where attack rate data is not incorporated, adding more data

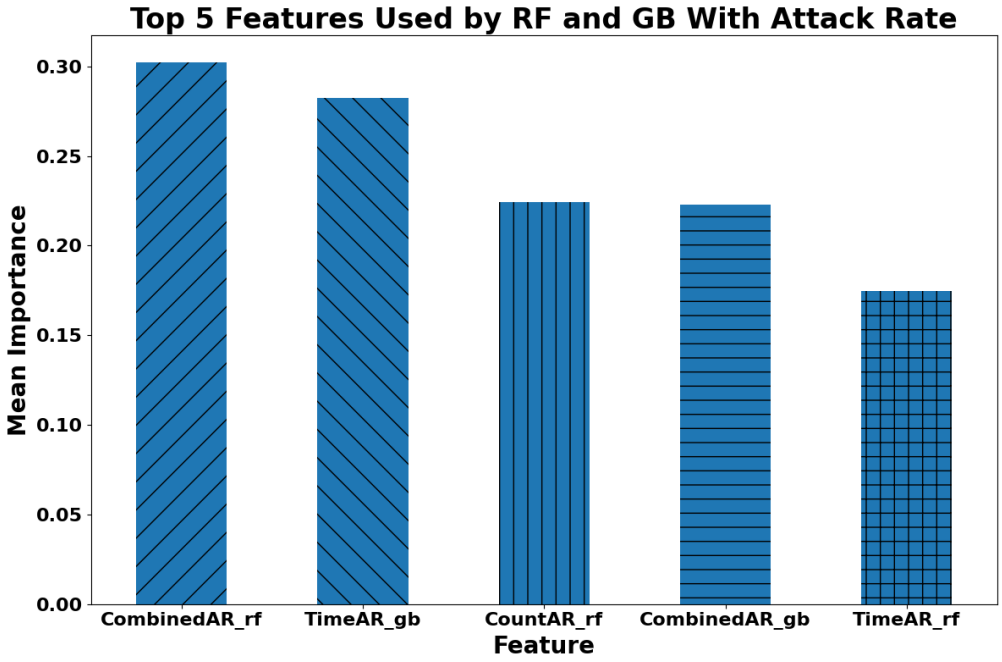


Figure 6.5: Significance of feature variables in gradient boosting and random forest models for predicting unprepared-for attacks: highlighting the role of calculated trust based on different attack rates from algorithm 6.1 (i.e., Count, Time, Combined)

does not substantially improve the predictive outcomes. Insights from a preceding experiment, 6.5.3, show this phenomenon, highlighting that the features most likely influencing prediction accuracy are related to the vehicle's position and speed alterations. Such attributes remain largely uninfluenced by an increase in honeypots, as these devices tend to focus on localized data collection. However, when attack rate information is integrated, it provides a deeper understanding of the attacker's behavior through their malicious engagements across the network, thereby enhancing the model's ability to predict attacks more accurately.

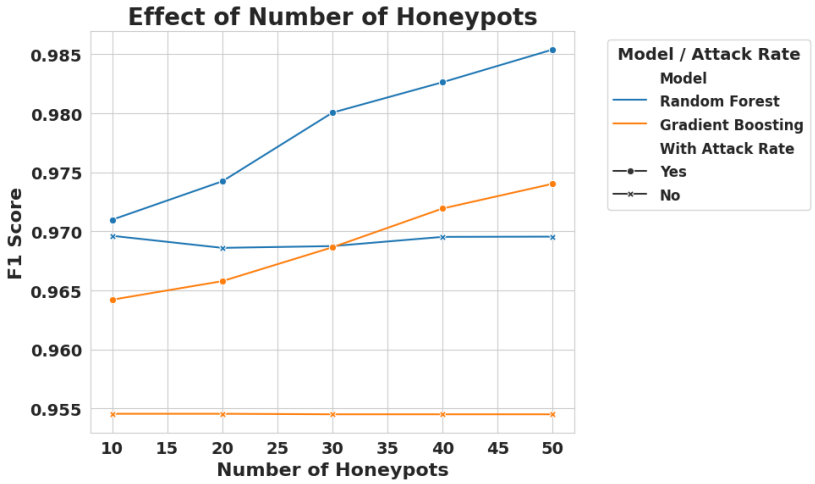


Figure 6.6: Improvement in F1 scores for random forest and gradient boosting models with the use of attack rate data in predicting unprepared-for attacks

6.5.5 Honeypot Selection Strategies on Attack Prediction Accuracy

In this experiment, we aim to assess the impact of various honeypot selection strategies on the precision of the unprepared-for attack prediction capabilities within VANETs. This work utilizes data harvested from ten selected honeypots strategically deployed from a more extensive network-wide distribution. The experimental design is based on two pivotal considerations: firstly, the intention to streamline the data acquisition process by intentionally limiting the number of operational honeypots, thereby enhancing the efficiency of data collection, and secondly, to exploit a distinctive feature of VANETs, namely, the ability to map and understand the spatial distribution of network density. This feature facilitates the identification of zones with high vehicular activity, representing higher data density, thereby enabling a targeted selection of honeypot vehicles situated within these densely populated areas.

The main aim of this experiment is to compare two distinct honeypot selection methodologies: one approach involves the random selection of honeypots without any

specific criteria or systematic consideration; the alternative method employs a more deliberate strategy, selecting honeypots based on their placement within areas characterized by high network density. Through this comparison, we seek to highlight the efficacy of each selection strategy in augmenting the accuracy of attack predictions, thereby contributing valuable insights into optimizing honeypot deployment for enhanced network security.

The results presented in Figures 6.7 and 6.8 compare the effectiveness of deploying randomly selected honeypots against that of employing honeypots characterized by a high data exchange rate, referred to as top honeypots. The figures comprehensively assess predictive performance metrics, including accuracy, precision, recall, and F1 score. These metrics are analyzed in the context of whether attack rate data is utilized and according to the honeypot selection strategy employed (random vs. top honeypots). This analytical exploration of the influence of honeypot selection on the accuracy of predictions shows a notable superiority in favor of selecting top honeypots characterized by high data exchange rates. This finding underscores the critical importance of strategic honeypot placement in enhancing the efficacy of network attack predictions.

6.6 Summary

In this chapter, we presented an often-overlooked aspect of the TMS, specifically the trust origin, by assessing its correlation with network attack rates to enhance trust evaluation and subsequent prediction of novel attacks. We developed a robust system to detect anticipated (prepared-for) and protect unforeseen (unprepared-for) attacks within network systems through a methodological analysis of attack rates associated

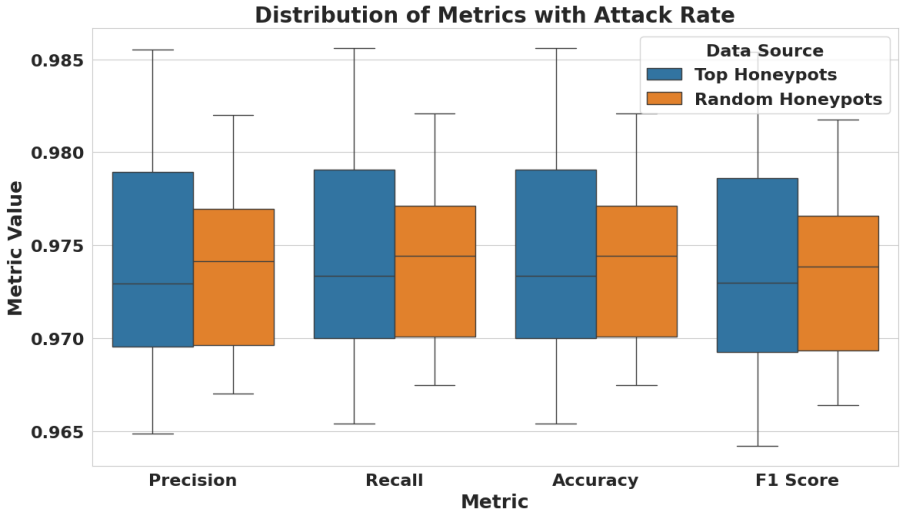


Figure 6.7: Performance metrics comparison using attack rate data with randomly selected vs. high data exchange rate (Top) honeypots, showcasing accuracy, precision, recall, and F1 score

with known attacks. Integrating ML algorithms with strategically placed honeypots, the system presents a holistic defense mechanism for VANETs. It excels in predicting known attacks and identifying and countering novel attacks by analyzing attack rates gleaned from honeypot data, thus reinforcing VANET security by focusing on attack interactions while upholding ethical considerations such as privacy. Our innovative algorithm calculates the trust level based on interactions within the network, utilizing dynamic honeypots to ensure widespread coverage and vehicular safety. The comprehensive evaluation demonstrated the efficacy of using attack rates for predicting previously unidentified attacks, improving detection accuracy and F1 scores without generating too many false positives, a common problem associated with honeypots. Furthermore, the research provides actionable insights into optimizing honeypot deployment for maximal security enhancement, illustrating the significant advantages of our approach to strengthening VANET protection against a broad spectrum of

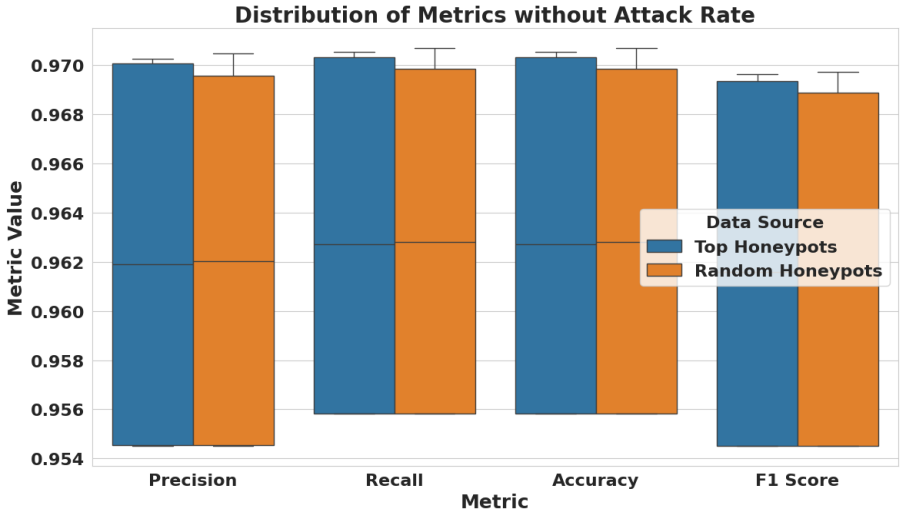


Figure 6.8: Effectiveness of honeypot selection strategies without attack rate data, analyzing predictive performance metrics (accuracy, precision, recall, F1 score) between random and top honeypots

attacks.

Chapter 7

Conclusions

This thesis addresses implementing proactive security measures in Vehicular ad-hoc Networks (VANETs), which will help improve road safety and neutralize cyber threats that can compromise human safety and network efficacy. Our proposed strategies target the ad-hoc segments in VANETs, which are particularly vulnerable to insider attacks. Trust Management Systems (TMS) have proven effective against post-authenticated nodes and comprise three components: trust subject, trust service, and trust origin. Our proposed proactive security approaches have been applied to these components to enhance overall system security.

7.1 Summary

Chapter 1 provided an overview of the research problem and outlined our contributions to the field of VANET security. Chapter 2 delved into the background and literature, exploring the significance, foundational technologies, and key principles of VANET security, including aspects of attack prediction and trust management systems.

In Chapter 3, we advanced the trust subject component of TMS by employing

Situational Awareness (SA) to evaluate the trustworthiness of entities. Utilizing the publicly available VeReMi dataset, we demonstrated that incorporating SA with Machine Learning (ML) models significantly improves the detection of misbehavior attacks. Our findings indicate that SA enhances the network's sensitivity to detect misbehaving nodes, resulting in a more than 50% increase in recall in some scenarios, underscoring the potential of SA-enhanced ML models in securing VANETs by enabling earlier and more accurate threat identification. The proposed approach effectively alerts trust subjects to the consequences of sharing false data, thus discouraging such behavior and enhancing overall network reliability.

In Chapter 4, we addressed the limitations of existing datasets by extending the VeReMi dataset using F2MD and incorporating Cooperative Awareness Messages (CAMs) and False Reporting Attacks (FRA). Our enhanced dataset closely mimics real-world conditions and vulnerabilities in VANETs. Through simulations using Recurrent Neural Network (RNN) models, including a stacked model of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), we were able to predict attack endgames up to several minutes in advance. This work emphasizes the importance of developing countermeasures to mitigate these effects and enhance VANET security.

Chapter 5 targeted the trust service component by predicting and mitigating the effects of attacks. Building on the work established in Chapter 4, we extended our predictive approach to consider the cascading effects of such attacks. We introduced an algorithmic model designed to predict and effectively mitigate the impact of attack endgames. Our approach includes a three-layered geofencing method using DBSCAN clustering and the convex hull algorithm to manage communication within high-traffic

areas efficiently. Additionally, we introduced a blacklisting technique for the removal of misbehaving vehicles. Simulation results demonstrate that the proposed approach effectively mitigates the impact of attacks, including reducing hazards and restoring vehicle speed recovery.

In Chapter 6, we presented the trust origin component by assessing its correlation with network attack rates to enhance trust evaluation and predict unprepared-for attacks. We developed a system model that actively uses honeypots to calculate attack rates and identify threats. By facilitating an attack prediction mechanism, the system extrapolates attack rates from prepared-for attacks to anticipate unprepared-for attacks. Additionally, we proposed dynamic honeypots tailored to the evolving VANET environment, which operate passively to avoid disseminating false messages and focus on interaction patterns without compromising vehicle identities. Simulations demonstrate the system's capability to predict unprepared-for attacks without increasing false positives and showcase the optimal use of honeypots for maximum efficiency.

7.2 Limitations

While the proposed methods have demonstrated substantial advancements in VANET security, they are not without limitations. One limitation is the absence of real-life implementation of VANETs with a sufficient number of deployed vehicles. This lack of data hindered our ability to fully observe the real-world impact of ongoing attacks and understand user behavior in response to these threats related to the attack endgame manifestation.

A second significant limitation is the uncontrollable imbalance of malicious activities and attack messages when generating data sets. By nature, malicious activities are far less than normal ones. However, Intrusion Detection Systems (IDSs) require an adequate representation of malicious activities to train effectively and identify threats accurately. An overabundance of such activities can lead to the normalization of these behaviors by the IDSs, thereby diminishing their effectiveness. Balancing the quantity and variety of malicious activities in training datasets is crucial to maintaining the IDSs' sensitivity and reliability in detecting genuine threats.

7.3 Future Work

This thesis has presented an approach for implementing proactive security in VANETs. The presented approaches achieve prediction and forecasting for upcoming threats and malicious users, giving network administrators or vehicles a reasonable amount of time to respond. However, open challenges and potential implementation opportunities can still be investigated to enhance the proposed approaches. A few of these avenues are listed below.

7.3.1 Proactive Security on the Infrastructure Level

Implementing proactive security on the ad-hoc level, as presented in this thesis, showed its advantages when paired with TMS. However, proactive security implementation on the infrastructure level still needs to be pursued. The infrastructure level involves multiple entities, such as trusted authorities, third parties, or service providers. Research is needed to determine where to apply the proactive security approach, considering potential regulatory challenges.

7.3.2 Proactive Security to Mitigate VANET Physical Damage

VANET attacks that can cause physical damage are becoming increasingly popular among hackers. Examples of these attacks include remotely starting or stopping the vehicle or controlling some parts of the vehicle. Such attacks can have disastrous effects on a moving vehicle. Although these attacks can be addressed at the network level, datasets that include physical world data are needed to apply the proposed attack endgame prediction and mitigation approach. Developing a dataset that includes network and CAN bus data in smart vehicles can link the network and physical layers. With such data, we can introduce countermeasures on the physical layer if these attacks cannot be stopped at the network level.

7.3.3 Cross-Regional Reputation Sharing

The proactive security approach considers sharing reputation and trust scores within the same VANET in a limited geographical area. However, sharing reputation or trust scores for TMS between geographically distanced areas presents a challenge. This capability is essential given vehicular mobility, thus necessitating a seamless transition and continuity in trust evaluations to maintain network security and reliability. Sharing reputation scores across different areas would ensure that vehicles do not lose their accumulated trustworthiness when entering a different geographic region, thus preventing potential exploitation by malicious entities that target vehicles new to the area. However, this task is fraught with challenges, including ensuring the consistency and integrity of reputation data during transfer, handling the potential increase in communication overhead, and addressing privacy concerns. Additionally, implementing a standardized protocol for reputation sharing that is secure and efficient across

various administrative domains adds another layer of complexity.

7.4 Concluding Remarks and Recommendations

Our findings demonstrate that there are numerous areas for further research and development in VANET security, as well as several key issues that need resolution for future researchers.

Firstly, security prediction and forecasting are pivotal for enhancing VANET security. More research is needed in this area to develop robust methods for attack prediction. The increasing ability to collect data from various sources, coupled with advanced processing techniques, offers the potential to uncover hidden patterns that may indicate impending or ongoing attacks. Nevertheless, a significant challenge still exists, which is the requirement for comprehensive datasets. There is a notable deficiency in datasets for evaluating security prediction techniques and real-world data from actual VANET attacks. This deficiency must be addressed so that researchers can advance the field effectively.

Secondly, regulations concerning VANETs are still evolving. Additionally, standards regarding VANETs implementation differ and might change or be waiting for government approval. It is imperative for research in this domain to stay current with newly established regulations or changes in standards. As the regulatory landscape continues to develop, it will influence the design and implementation of security measures in VANETs. Researchers must remain adaptable and informed about these changes to ensure compliance and relevance.

Thirdly, real-life testing of VANET security solutions presents substantial challenges. While simulations have improved in replicating real-world conditions, they

cannot fully capture the complexities and unpredictability of human behavior. Real-world validation assesses the effectiveness of security solutions against factors such as non-compliance with traffic laws, erratic driving, varied reactions to system alerts, differences in technology adoption, and psychological responses to warnings. Therefore, expanding opportunities for real-world testing is needed.

Additionally, a persistent issue in using Machine Learning (ML) for security research is the imbalance between benign and malicious activities. In VANETs, this problem is exacerbated by the short time window available to detect certain attacks before they pose a threat to human safety. Ensuring that ML models are trained on sufficiently diverse and representative datasets is critical to improving their accuracy and reliability in detecting malicious activities promptly.

Lastly, the industry representation for VANET security is limited. Few companies and organizations are actively working on VANET security implementation, which underscores the need for more government, automobile industry, and academia involvement to establish a collaboration that fosters innovation and discovers practical applications for research findings.

Bibliography

[1] M. Alam, J. Ferreira, and J. Fonseca, "Introduction to intelligent transportation systems," in *Intelligent Transportation Systems*. Springer, 2016, pp. 1–17.

[2] "Intelligent Transportation System Market worth $68.0 billion by 2026 - Exclusive Report by MarketsandMarkets [Online]," Available: https://www.prnewswire.com/news-releases/intelligent-transportation-system-market-worth-68-0-billion-by-2026--exclusive-report-by-marketsandmarkets-301414119.html.

[3] Y. Yang and R. Bagrodia, "Evaluation of VANET-based advanced intelligent transportation systems," in *Proceedings of the Sixth ACM International Workshop on VehiculAr InterNETworking*, 2009, p. 3–12.

[4] M. Gerla, E.-K. Lee, G. Pau, and U. Lee, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular clouds," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 2014, pp. 241–246.

[5] T. Bécsi, S. Aradi, and P. Gáspár, "Security issues and vulnerabilities in connected car systems," in *International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2015, pp. 477–482.

[6] T. Halabi, O. A. Wahab, R. Al Mallah, and M. Zulkernine, "Protecting the internet of vehicles against advanced persistent threats: A bayesian stackelberg game," *IEEE Transactions on Reliability*, pp. 970–985, 2021.

[7] Y. Zang, L. Stibor, H.-J. Reumerman, and H. Chen, "Wireless local danger warning using inter-vehicle communications in highway scenarios," in *14th European Wireless Conference*, 2008, pp. 1–7.

[8] O. Ben Fredj, A. Mihoub, M. Krichen, O. Cheikhrouhou, and A. Derhab, "Cybersecurity attack prediction: a deep learning approach," in *13th International Conference on Security of Information and Networks*, 2020, pp. 1–6.

[9] S. Ftaimi and T. Mazri, "A comparative study of machine learning algorithms for VANET networks," in *Proceedings of the 3rd International Conference on Networking, Information Systems & Security*, 2020, pp. 1–8.

[10] S. B. Raut and L. Malik, "Survey on vehicle collision prediction in VANET," in *2014 IEEE International conference on computational intelligence and computing research*, 2014, pp. 1–5.

[11] H. Al-Khateeb, G. Epiphaniou, A. Reviczky, P. Karadimas, and H. Heidari, "Proactive threat detection for connected cars using recursive Bayesian estimation," *IEEE Sensors Journal*, pp. 4822–4831, 2017.

[12] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "Prediction-based GNSS spoofing attack detection for autonomous vehicles," *arXiv preprint arXiv:2010.11722*, 2020.

[13] Z. Nezhadian, E. Branca, and N. Stakhanova, "Analysis and prediction of web proxies misbehavior," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, ser. ARES '22, 2022.

[14] M. A. Abdelmaguid, H. S. Hassanein, and M. Zulkernine, "SAMM: Situation awareness with machine learning for misbehavior detection in VANET," in *Proceedings of the 17th International Conference on Availability, Reliability and Security*, 2022, pp. 1–10.

[15] M. R. Endsley, "Measurement of situation awareness in dynamic systems," *Human Factors: The Journal of Human Factors and Ergonomics Society*, pp. 65 – 84, 1995.

[16] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in VANETs," *International Conference on Security and Privacy in Communication Systems*, pp. 318–337, 2018.

[17] M. A. Abdelmaguid, H. S. Hassanein, and M. Zulkernine, "Attack endgame: Proactive security approach for predicting attack consequences in VANET," in *ICC IEEE International Conference on Communications*, 2023, pp. 3762–3767.

[18] ——, "A VeReMi-based dataset for predicting the effect of attacks in VANETs," in *Proceedings of the Int'l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, 2023, p. 175–179.

[19] J. Liu and A. J. Khattak, "Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles," *Transportation Research Part C: Emerging Technologies*, pp. 83–100, 2016.

[20] N. Lyamin, A. Vinel, M. Jonsson, and B. Bellalta, "Cooperative awareness in VANETs: On ETSI EN 302 637-2 performance," *IEEE Transactions on Vehicular Technology*, pp. 17–28, 2018.

[21] IEEE, "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE Std 1609.2-2006*, pp. 1–105, 2006.

[22] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proceedings of the IEEE*, pp. 1162–1182, 2011.

[23] T. Halabi and M. Zulkernine, "Reliability-driven task assignment in vehicular crowdsourcing: A matching game," in *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2019, pp. 78–85.

[24] C. Zoghlami, R. Kacimi, and R. Dhaou, "5G-enabled V2X communications for vulnerable road users safety applications: a review," *Wireless Networks*, pp. 1237–1267, 2023.

[25] Z. Xu, X. Li, X. Zhao, M. H. Zhang, Z. Wang *et al.*, "DSRC versus 4G-LTE for connected vehicle applications: A study on field experiments of vehicular communication performance," *Journal of Advanced Transportation*, 2017.

[26] J. Sun, S. Iqbal, N. Seifollahpour Arabi, and M. Zulkernine, "A classification of attacks to in-vehicle components (IVCs)," *Vehicular Communications*, p. 100253, 2020.

[27] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular Ad-hoc Networks (VANETS): status, results, and challenges," *Telecommunication Systems*, pp. 217–241, 2012.

[28] S. Khan, I. Sharma, M. Aslam, M. Z. Khan, and S. Khan, "Security challenges of location privacy in VANETs and state-of-the-art solutions: A survey," *Future Internet*, vol. 13, 2021.

[29] C. Ishida, Y. Arakawa, I. Sasase, and K. Takemori, "Forecast techniques for predicting increase or decrease of attacks using bayesian inference," in *PACRIM. IEEE Pacific Rim Conference on Communications, Computers and signal Processing*, 2005, pp. 450–453.

[30] D. H. Kim, T. Lee, S.-O. D. Jung, H. P. In, and H. J. Lee, "Cyber threat trend analysis model using HMM," in *Third International Symposium on Information Assurance and Security*, 2007, pp. 177–182.

[31] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Transactions on Information Forensics and Security*, pp. 1775–1789, 2013.

[32] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: ARIMA vs. LSTM," *ArXiv*, 2018.

[33] X. Fang, M. Xu, S. Xu, and P. Zhao, "A deep learning framework for predicting cyber attacks rates," *EURASIP Journal on Information Security*, p. 5, 2019.

[34] E. Gandotra, D. Bansal, and S. Sofat, "Computational techniques for predicting cyber threats," in *Intelligent Computing, Communication and Devices*, 2015, pp. 247–253.

[35] M. Prasath and B. Perumal, "Network attack prediction by random forest: Classification method," in *3rd International conference on Electronics, Communication and Aerospace Technology*, 2019, pp. 647–654.

[36] S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, and B. Yang, "Predicting network attack patterns in SDN using machine learning approach," in *IEEE Conference on Network Function Virtualization and Software Defined Networks*, 2016, pp. 167–172.

[37] A. Azaria, A. Richardson, S. Kraus, and V. Subrahmanian, "Behavioral analysis of insider threat: A survey and bootstrapped prediction in imbalanced data," *IEEE Transactions on Computational Social Systems*, pp. 135–155, 2014.

[38] M. Al-Qurishi, M. Alrubaian, S. M. M. Rahman, A. Alamri, and M. M. Hassan, "A prediction system of Sybil attack in social network using deep-regression model," *Future Generation Computer Systems*, pp. 743–753, 2018.

[39] H. Schafer, E. Santana, A. Haden, and R. Biasini, "A commute in data: The Comma2k19 dataset," *arXiv arXiv:1812.05752*, 2018.

[40] D. S. Lavrova, E. Popova, and A. Shtyrkina, "Prevention of DoS attacks by predicting the values of correlation network traffic parameters," *Automatic Control and Computer Sciences*, pp. 1065–1071, 2019.

[41] M. Husák, J. Komárková, E. Bou-Harb, and P. Čeleda, "Survey of attack projection, prediction, and forecasting in cyber security," *IEEE Communications Surveys & Tutorials*, pp. 640–660, 2018.

[42] P. Gao, B. Wang, N. Z. Gong, S. R. Kulkarni, K. Thomas, and P. Mittal, "Sybilfuse: Combining local attributes with global structure to perform robust sybil detection," in *IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[43] S. O. Uwagbole, W. J. Buchanan, and L. Fan, "Applied machine learning predictive analytics to SQL injection attack detection and prevention," in *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017, pp. 1087–1090.

[44] A. Okutan, S. J. Yang, and K. McConky, "Predicting cyber attacks with Bayesian networks using unconventional signals," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, 2017, pp. 1–4.

[45] G. Onoh, "Predicting cyber-attacks using publicly available data," in *Journal of The Colloquium for Information System Security Education*, vol. 6, no. 1, 2018, pp. 18–18.

[46] R. Hussain, J. Lee, and S. Zeadally, "Trust in VANET: A survey of current solutions and future research opportunities," *IEEE Transactions on Intelligent Transportation Systems*, pp. 2553–2571, 2020.

[47] Y.-B. Leau and S. Manickam, "Network security situation prediction: a review and discussion," in *International Conference on Soft Computing, Intelligence Systems, and Information Technology*, 2015, pp. 424–435.

[48] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 2008, pp. 1238–1246.

[49] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys & Tutorials*, pp. 779–811, 2018.

[50] N. Bißmeyer, J. Njeukam, J. Petit, and K. M. Bayarou, "Central misbehavior evaluation for vanets based on mobility data plausibility," in *Proceedings of the Ninth ACM International Workshop on Vehicular Inter-Networking, Systems, and Applications*, 2012, p. 73–82.

[51] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in VANET," pp. 644–653, 2011.

[52] M. Abu-Elkheir, S. A. Hamid, H. S. Hassanein, I. M. Elhenawy, and S. Elmougy, "Position verification for vehicular networks via analyzing two-hop neighbors information," in *2011 IEEE 36th Conference on Local Computer Networks*, 2011, pp. 805–812.

[53] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[54] S. Gyawali, Y. Qian, and R. Q. Hu, "Machine learning and reputation based misbehavior detection in vehicular communication networks," *IEEE Transactions on Vehicular Technology*, pp. 8871–8885, 2020.

[55] A. Gruebler, K. D. McDonald-Maier, and K. M. A. Alheeti, "An intrusion detection system against black hole attacks on the communication network of self-driving cars," in *2015 Sixth International Conference on Emerging Security Technologies (EST)*, 2015, pp. 86–91.

[56] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in VANET," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 564–571.

[57] Q. Li, A. Malip, K. M. Martin, S.-L. Ng, and J. Zhang, "A reputation-based announcement scheme for VANETs," *IEEE Transactions on Vehicular Technology*, pp. 4095–4108, 2012.

[58] X. Hong, D. Huang, M. Gerla, and Z. Cao, "SAT: situation-aware trust architecture for vehicular networks," pp. 31–36, 2008.

[59] Y. Jiang, C.-h. Li, L.-s. Yu, and B. Bao, "On network security situation prediction based on RBF neural network," pp. 4060–4063, 2017.

[60] M. H. Eiza, T. Owens, Q. Ni, and Q. Shi, "Situation-aware QoS routing algorithm for vehicular ad hoc networks," *IEEE Transactions on vehicular technology*, pp. 5520–5535, 2015.

[61] R. Mitchell and I.-R. Chen, "A survey of intrusion detection techniques for cyber-physical systems," *ACM Computing Surveys (CSUR)*, pp. 1–29, 2014.

[62] S. So, J. Petit, and D. Starobinski, "Physical layer plausibility checks for misbehavior detection in v2x networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, p. 84–93.

[63] S. Ahmed and K. Tepe, "Misbehaviour detection in vehicular networks using logistic trust," in *2016 IEEE Wireless Communications and Networking Conference*, 2016, pp. 1–6.

[64] S. J. Selcon and R. Taylor, "Evaluation of the situational awareness rating technique (SART) as a tool for aircrew systems design," *AGARD, Situational Awareness in Aerospace Operations*, pp. 23–53, 1990.

[65] C. Onwubiko, "Understanding cyber situation awareness." *Int. J. Cyber Situational Aware.*, pp. 11–30, 2016.

[66] N. Elmrabit, F. Zhou, F. Li, and H. Zhou, "Evaluation of machine learning algorithms for anomaly detection," in *2020 International Conference on Cyber Security and Protection of Digital Services*, 2020, pp. 1–8.

[67] A. Uprety, D. B. Rawat, and J. Li, "Privacy preserving misbehavior detection in IoV using federated machine learning," in *2021 IEEE 18th Annual Consumer Communications Networking Conference (CCNC)*, 2021, pp. 1–6.

[68] X.-X. Lin, P. Lin, and E.-H. Yeh, "Anomaly detection/prediction for the Internet of Things: State of the art and the future," *IEEE Network*, pp. 212–218, 2021.

[69] A. Wang, A. Mohaisen, and S. Chen, "An adversary-centric behavior modeling of DDoS attacks," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017, pp. 1126–1136.

[70] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Transactions on Vehicular Technology*, pp. 6631–6643, 2020.

[71] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "VeReMi extension: A dataset for comparable evaluation of misbehavior detection in VANETs," in *IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[72] D. Swessi and H. Idoudi, "A comparative review of security threats datasets for vehicular networks," in *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2021, pp. 746–751.

[73] U.S. Department of Transportation Federal Highway Administration, "Next generation simulation (NGSIM) vehicle trajectories and supporting data," Provided by ITS DataHub through Data.transportation.gov, 2016, accessed: 2023-07-06. [Online]. Available: http://doi.org/10.21949/1504477

[74] B. Coifman and L. Li, "A critical evaluation of the next generation simulation (NGSIM) vehicle trajectory dataset," *Transportation Research Part B: Methodological*, pp. 362–377, 2017.

[75] F. A. Ghaleb, A. Zainal, M. A. Rassam, and F. Mohammed, "An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications," in *2017 IEEE Conference on Application, Information and Network Security (AINS)*, 2017, pp. 13–18.

[76] M. Ingham, J. Marchang, and D. Bhowmik, "IoT security vulnerabilities and predictive signal jamming attack analysis in LoRaWAN," *IET Information Security*, pp. 368–379, 2020.

[77] S. C. HPL, "Introduction to the controller area network (CAN)," *Application Report SLOA101*, pp. 1–17, 2002.

[78] C. J. Hegarty and E. Chatre, "Evolution of the global navigation satellitesystem (GNSS)," *Proceedings of the IEEE*, pp. 1902–1917, 2008.

[79] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *International Journal of Signal Processing Systems*, pp. 256–262, 2013.

[80] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO-simulation of urban mobility," *International Journal on Advances in Systems and Measurements*, pp. 128–138, 2012.

[81] C. Sommer, D. Eckhoff, A. Brummer, D. Buse, F. Hagenauer, S. Joerer, M. Segata, A. Virdis, and M. Kirsche, "Recent advances in network simulation: The omnet++ environment and its ecosystem," pp. 55–106, 2019.

[82] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An interface for coupling road traffic and network simulators." Association for Computing Machinery, 2008, p. 155–163.

[83] L. Codecà, R. Frank, and T. Engel, "LuST: a 24-hour scenario of luxembourg city for SUMO traffic simulations," in *SUMO User Conference 2015 - Intermodal Simulation for Intermodal Transport*, 2015.

[84] K. Sjoberg, P. Andres, T. Buburuzan, and A. Brakemeier, "Cooperative intelligent transport systems in Europe: Current deployment status and outlook," *IEEE Vehicular Technology Magazine*, pp. 89–97, 2017.

[85] "IEEE Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *in IEEE Std 802.11-2012 (Revision of IEEE Std 802.11-2007)*, pp. 1–2793, 2012.

[86] Y. Yu, X. Zeng, X. Xue, and J. Ma, "LSTM-based intrusion detection system for VANETs: A time series classification approach to false message detection," *IEEE Transactions on Intelligent Transportation Systems*, pp. 23 906–23 918, 2022.

[87] P. T. Yamak, L. Yujian, and P. K. Gadosey, "A comparison between arima, lstm, and gru for time series forecasting," in *Proceedings of the 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 2019, pp. 49–55.

[88] A. K. Malhi, S. Batra, and H. S. Pannu, "Security of vehicular ad-hoc networks: A comprehensive survey," *Computers & Security*, p. 101664, 2020.

[89] Y. Hao, Y. Cheng, C. Zhou, and W. Song, "A distributed key management framework with cooperative message authentication in VANETs," *IEEE Journal on selected areas in communications*, pp. 616–629, 2011.

[90] H. Amari, Z. A. E. Houda, L. Khoukhi, and L. H. Belguith, "Trust management in vehicular ad-hoc networks: Extensive survey," *IEEE Access*, pp. 47 659–47 680, 2023.

[91] Y. Liang, E. Luo, and Y. Liu, "Physically secure and conditional-privacy authenticated key agreement for VANETs," *IEEE Transactions on Vehicular Technology*, pp. 7914–7925, 2023.

[92] S. Biswas, M. M. Haque, and J. V. Misic, "Privacy and anonymity in VANETs: A contemporary study." *Ad Hoc Sens. Wirel. Networks*, pp. 177–192, 2010.

[93] X. Wang, H. Zhang, T. Q. Duong, M. Elkashlan, and V. N. Q. Bao, "Secure cooperative communication with nth best relay selection," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, pp. 1–5.

[94] O. El Joubari, J. B. Othman, and V. Vèque, "TA-TDMA: A traffic aware TDMA MAC protocol for safety applications in VANET," in *2021 IEEE Symposium on Computers and Communications (ISCC)*, 2021, pp. 1–8.

[95] A. Sharma and A. Jaekel, "Machine learning based misbehaviour detection in VANET using consecutive BSM approach," *IEEE Open Journal of Vehicular Technology*, pp. 1–14, 2021.

[96] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, p. 132306, 2020.

[97] C. Kumar and S. Mohanty, "Current trends in cyber security for drones," in *International Carnahan Conference on Security Technology (ICCST)*, 2021, pp. 1–5.

[98] Q. Liu, S. Yan, J. Wan, R. Ma, and J. Xu, "Multi-malicious nodes double-spending attack blacklist management model," in *4th International Conference on Data Science and Information Technology*, 2021, pp. 462–467.

[99] G. Dong, Y. Jin, S. Wang, W. Li, Z. Tao, and S. Guo, "DB-Kmeans:An Intrusion Detection Algorithm Based on DBSCAN and K-means," in *20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2019, pp. 1–4.

[100] J. Cui, X. Zhang, H. Zhong, Z. Ying, and L. Liu, "RSMA: Reputation system-based lightweight message authentication framework and protocol for 5G-enabled vehicular networks," *IEEE Internet of Things Journal*, pp. 6417–6428, 2019.

[101] M. A. Al-Shareeda and S. Manickam, "A systematic literature review on security of vehicular ad-hoc network (VANET) based on veins framework," *IEEE Access*, 2023.

[102] H. Hewamalage, C. Bergmeir, and K. Bandara, "Recurrent neural networks for time series forecasting: Current status and future directions," *International Journal of Forecasting*, pp. 388–427, 2021.

[103] K. K. Kommineni and A. Prasad, "A review on privacy and security improvement mechanisms in MANETs," *International Journal of Intelligent Systems and Applications in Engineering*, pp. 90–99, 2024.

[104] G. D. Putra and S. Sulistyo, "Trust based approach in adjacent vehicles to mitigate sybil attacks in VANET," in *Proceedings of the International Conference on Software and E-Business*, 2017, p. 117–122.

[105] R. K. Dhanaraj, S. H. Islam, and V. Rajasekar, "A cryptographic paradigm to detect and mitigate blackhole attack in VANET environments," *Wirel. Netw.*, p. 3127–3142, 2022.

[106] Y. Zhuang, J. Pan, Y. Luo, and L. Cai, "Time and location-critical emergency message dissemination for vehicular ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, pp. 187–196, 2010.

[107] M. Tsukada, S. Arii, H. Ochiai, and H. Esaki, "Misbehavior detection using collective perception under privacy considerations," in *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, p. 808–814.

[108] S. Ercan, M. Ayaida, and N. Messai, "Misbehavior detection for position falsification attacks in VANETs using machine learning," *IEEE Access*, pp. 1893–1904, 2022.

[109] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled internet of vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Transactions on Vehicular Technology*, pp. 2906–2920, 2019.

[110] T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, "DeepADV: A deep neural network framework for anomaly detection in VANETs," *IEEE Transactions on Vehicular Technology*, pp. 12 013–12 023, 2021.

[111] M. Maad Hamdi, L. Audah, S. Abduljabbar Rashid, A. Hamid Mohammed, S. Alani, and A. Shamil Mustafa, "A review of applications, characteristics and challenges in vehicular ad hoc networks (VANETs)," in *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, 2020, pp. 1–7.

[112] I. A. Sumra, I. Ahmad, H. Hasbullah *et al.*, "Behavior of attacker and some new possible attacks in vehicular ad hoc network (VANET)," in *2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2011, pp. 1–8.

[113] R. Al-Ani, T. Baker, B. Zhou, and Q. Shi, "Privacy and safety improvement of VANET data via a safety-related privacy scheme," *International Journal of Information Security*, pp. 1–21, 2023.

[114] H. Laghbi, S. Alateef, and N. Thomas, "The impact of resource DoS and propagation loss on VANET routing," in *39 th Annual UK Performance Engineering Workshop*, 2023, p. 50.

[115] S. A. Asra, "Security issues of vehicular ad hoc networks (VANET): A systematic review," *TIERS Information Technology Journal*, pp. 17–27, 2022.

[116] I. Elleuch, A. Makni, and R. Bouaziz, "Towards a cooperative intelligent system for unpredictable and predictable road hazard detection," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl*, pp. 101–114, 2019.

[117] ——, "Design of an intelligent cooperative road hazard detection persistent system," in *Intelligent Systems Design and Applications*, 2020, pp. 696–707.

[118] E. C. Thompson, *Cybersecurity incident response: How to contain, eradicate, and recover from incidents*. Apress, 2018.

[119] N. Mhaskar, M. Alabbad, and R. Khedri, "A formal approach to network segmentation," *Computers & Security*, p. 102162, 2021.

[120] J. M. Kizza, "Firewalls," in *Guide to Computer Network Security*, 2024, pp. 265–294.

[121] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, pp. 1–22, 2019.

[122] S. Shivshankar and A. Jamalipour, "An evolutionary game theory-based approach to cooperation in VANETs under different network conditions," *IEEE Transactions on Vehicular Technology*, pp. 2015–2022, 2014.

[123] Z. El-Rewini, K. Sadatsharan, D. F. Selvaraj, S. J. Plathottam, and P. Ranganathan, "Cybersecurity challenges in vehicular communications," *Vehicular Communications*, p. 100214, 2020.

[124] S. Kudva, S. Badsha, S. Sengupta, H. La, I. Khalil, and M. Atiquzzaman, "A scalable blockchain based trust management in VANET routing protocol," *Journal of Parallel and Distributed Computing*, pp. 144–156, 2021.

[125] A. S. Al Hasan, M. S. Hossain, and M. Atiquzzaman, "Security threats in vehicular ad hoc networks," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2016, pp. 404–411.

[126] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Transactions on Information Forensics and Security*, pp. 1666–1677, 2015.

[127] ——, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Transactions on Information Forensics and Security*, pp. 1775–1789, 2013.

[128] M. L. Bringer, C. A. Chelmecki, and H. Fujinoki, "A survey: Recent advances and future trends in honeypot research," *International Journal of Computer Network and Information Security*, p. 63, 2012.

[129] P. Patel and R. Jhaveri, "A honeypot scheme to detect selfish vehicles in vehicular ad-hoc network," in *Computing and Network Sustainability: Proceedings of IRSCNS 2016*, 2017, pp. 389–401.

[130] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in VANETs and VANET cloud," *Vehicular communications*, pp. 138–164, 2018.

[131] L. Zobal, D. Kolář, and R. Fujdiak, "Current state of honeypots and deception strategies in cybersecurity," in *11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2019, pp. 1–9.

[132] P. A. Pavlou, "Consumer acceptance of electronic commerce: Integrating trust and risk with the technology acceptance model," *International journal of electronic commerce*, pp. 101–134, 2003.

[133] A. Fabi and S. M. Thampi, "A trust management framework using forest fire model to propagate emergency messages in the internet of vehicles (IoV)," *Vehicular Communications*, vol. 33, p. 100404, 2022.

[134] J. Zhang, "A survey on trust management for VANETs," in *IEEE International Conference on Advanced Information Networking and Applications*, 2011, pp. 105–112.

[135] A. Hbaieb, S. Ayed, and L. Chaari, "A survey of trust management in the internet of vehicles," *Computer Networks*, p. 108558, 2022.

[136] T. Yagi, N. Tanimoto, T. Hariu, and M. Itoh, "Enhanced attack collection scheme on high-interaction web honeypots," in *The IEEE symposium on Computers and Communications*, 2010, pp. 81–86.

[137] R. C. Perkins and C. J. Howell, "Honeypots for cybercrime research," *Researching Cybercrimes: Methodologies, Ethics, and Critical Approaches*, pp. 233–261, 2021.