9-12-2024

Detecting and Understanding Position Falsification Attacks using Explainable Artificial Intelligence.

Mahesh Abburi
University of Windsor

Detecting and Understanding Position Falsification Attacks using Explainable Artificial Intelligence

By

Mahesh Abburi

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2024

Detecting and Understanding Position Falsification Attacks using Explainable Artificial Intelligence

by

Mahesh Abburi

APPROVED BY:

H. Wu
Department of Electrical & Computer Engineering

S. Jiang
School of Computer Science

A. Jaekel, Advisor
School of Computer Science

August 29, 2024

DECLARATION OF CO-AUTHORSHIP

I. Co-Authorship

I hereby declare that this thesis incorporates material that is the result of research conducted under the supervision of Dr. Arunita Jaekel. In all cases, the key ideas, primary contribution, experimental designs, data analysis, and interpretation were performed by the author, and the contribution of the co-author was primarily through providing feedback and the proofreading of the published manuscripts.

I am aware of the University of Windsor Senate Policy on Authorship, and I certify that I have properly acknowledged the contribution of other researchers to my thesis and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis. I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my work.

II. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owners) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Vehicular ad-hoc network (VANET) is an emerging technology for vehicle-to-vehicle communication vital for reducing road accidents and traffic congestion in an Intelligent Transportation System (ITS). Integrating Vehicular Ad-Hoc Networks (VANETs) into modern Intelligent Transportation Systems (ITS) has brought about significant advancements in transportation efficiency and safety. However, it has also introduced critical security concerns, particularly regarding the integrity of data exchanged among vehicles. This research focuses on tackling the emerging threat of Position Falsification Attacks in VANETs, where malicious entities broadcast fictitious location information to disrupt traffic flow and compromise road safety. Our methodology employs a detailed examination of the VeReMi dataset, a standard benchmark in VANETs security research, alongside state-of-the-art machine learning classification algorithms. A key focus is not only on developing robust detection models but also on integrating XAI to enhance the interpretability of the outcomes. This approach ensures that the underlying decision-making processes of the ML models are transparent and understandable, fostering trust and facilitating more accessible validation by human experts. Including XAI has demonstrated potential in providing deeper insights into model behaviours, particularly in understanding why specific predictions are made, thus identifying areas for model improvement. This thesis highlights the critical need to secure VANETs against Position Falsification Attacks and proposes an innovative solution by merging machine learning with explainable artificial intelligence. The findings contribute theoretically and practically, enhancing our understanding of VANET security challenges and providing actionable insights that can be implemented to safeguard vehicular communication networks against emerging cyber threats.

DEDICATION

I dedicate this thesis to my mother and father, brother, my friends for thier support and encouragement and also my supervisor for her guidance throughout my research

ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisor Dr. Arunitha Jaekel and PhD student Muhammad Anwar Shahid, for guidance, support and encouragement throughout my research. Moreover. I would like to thank my internal reader Dr. Shaoquan Jiang and external reader Dr. Huapeng Wu for thier support and feedback to improve my thesis. Finally, I would like to thank my parents and brother for providing the confidence and strength to complete my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

VANET	Vehicular ad-hoc network
RSU	Road-side Unit
OBU	On-board Unit
DSRC	Dedicated short range communication
C-V2X	Cellular Vehicle to everything
PKI	Public key infrastructure
BSM	Basic safety message
ITS	Intelligent Transportation system
WAVE	Wireless Access in vehicular Environment
VeReMi	Vehicular Reference Misbehaviour Dataset
XAI	Explainable Artificial Intelligence

CHAPTER 1

Introduction

1.1 Vehicular ad-hoc networks

Intelligent Transportation System [1] is an advanced technology that can improve road safety traffic management and reduce traffic congestion in the transportation system. According to the 2018 Global status report on road safety by the World Health Organisation (WHO), road accidents and injuries have become the 8th leading cause of death, with 1.35 million deaths annually. Moreover, road accidents are the 1st leading cause of death for children and young adults aged 5-29 years [2]. Vehicular ad hoc network (VANET)[3] is the emerging technology in the Intelligent Transportation System (ITS). Through information flow and communication, VANET can make the transportation network more efficient, secure, and safe. It is a highly dynamic wireless ad hoc network formed using vehicles, roadside units, and other infrastructures. As VANET has a rapidly changing topology and high mobility, and vehicles in the network can be stationary or continuously moving. Vehicles in the network are installed with On- Board Unit (OBU), which transmits a vehicle's status in the network to other nodes periodically. Road-side Units (RSU) are infrastructures stationed on the road's side, which provide services and help communication between the nodes in the network. There are other infrastructures, such as the Central Authority/ Authorization Party, which provides support such as registering a node in the network and revoking them in case of misbehaviour[4].

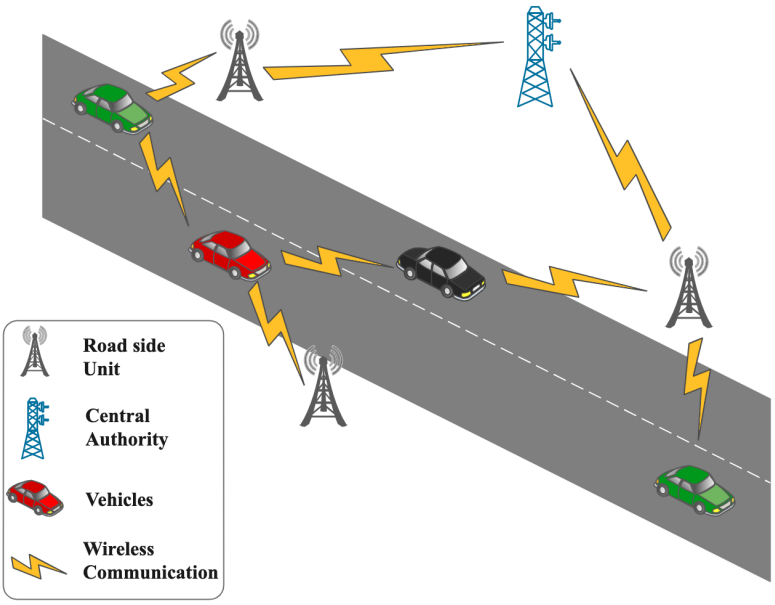

Figure 1.1: An example of Vehicular ad-hoc network [5]

In 1999, the Federal Communication Commission (FCC) of the United States allocated Dedicated Short-Range Communication (DSRC), a licensed spectrum of 75MHz in 5.9 GHz frequency bandwidth for communication between vehicles and road-side units [6]. DSRC is a service used for short to medium-range communication that provides high data transfer with minimum latency. Wireless Access in Vehicular Environment (WAVE) is the IEEE 1609 family standard protocol that uses the IEEE 802.11p standard to support communication in the vehicular network and provide standards for DSRC [7]. As DSRC has limitations in transferring a large amount of data and accessing the Internet of vehicles, a new standard is introduced, Cellular-V2X (C-V2X), which gives a better connectivity scope. C-V2X stands for the cellular vehicle to everything, and this cellular technology is designed to connect vehicles to other vehicles, roadside units, central authority and cloud-based services [8].

Communication in VANET is of different kinds, such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Infrastructure-to-Infrastructure (I2I) and Vehicle-to other devices (V2X). VANET supports two types of applications: Comfort applications and safety applications. Comfort application includes comfort-based communication such as weather information, advertisements, pricing and details about the

nearest gas stations or restaurants. However, the safety application includes safety-based communication between vehicles and infrastructures. Examples of safety applications are blind-spot warnings, emergency warnings, and lane change assistance. Wireless communication in the network can provide important information to the drivers or vehicles in time. However, wireless communication is vulnerable to various security and privacy attacks, which can cause misbehaviour in the network; hence, this information transmitted in the network must be verified and authenticated for correctness.

We can classify attackers in the network into the following [9]:

1. Insider vs. Outsider Attacker: Insider attackers are those who are authenticated members of the network, while outside attackers are those who are not authorized.

2. Active vs. Passive Attacker: Active attackers take part in the attack by directly interfering in the attack, such as altering the message or destroying the message packet in the network. Passive attackers listen to the conversation in the network without interfering directly and may use the information for malicious purposes.

3. Malicious vs. Rational Attacker: Attack that harms the network or causes extreme damage to the network by a malicious attacker. In comparison, rational attackers trigger the attack for personal gain

Vehicles in the VANET network send periodic status messages; such messages are called Basic Safety Messages (BSM). BSM contains the vehicle's current status, such as position coordinates, vehicle speed, and transmission time, which is broadcasted in the network periodically. These messages are digitally signed using cryptographic techniques [10] before broadcast, and only the authorized members of the network can access these BSMs.

VANET, being a wireless network, is susceptible to attacks and detecting these attacks can be termed as misbehaviour detection. Misbehaviour detection can be divided into

node-centric detection, where the detection of misbehaviour depends on the credibility of the node and data-centric detection, where detection is based on data reliability.

This thesis aims to detect a Position falsification attack, where the attacker vehicle in the network sends a false position coordinate in the BSM. Position falsification attacks can lead to traffic congestion and even accidents and cause severe damage to the network.

Five types of position falsification attacks detected in this research are:

1. Constant attack: Attacker vehicle transmits a fixed position in the network.
2. Constant offset attack: Attacker vehicle transmits a position with a fixed offset added to the actual position.
3. Random attack: Attacker vehicle transmits random position from the playground.
4. Random offset attack: Attacker vehicle transmits a uniformly random position from a pre-defined rectangle around the vehicle.
5. Eventual stop attack: Attacker vehicle behaves like a legitimate vehicle for some time and then transmits a current position repeatedly in the network.

Cryptographic techniques can provide message integrity but do not ensure message correctness; hence, cryptographic methods are insufficient to ensure network security. Additional detection methods are required to detect malicious vehicles sending false information in the network.

1.2 Motivation

The integration of Vehicular Ad-Hoc Networks (VANETs) into modern transportation systems has revolutionized road safety and efficiency, facilitating the exchange of critical safety information among vehicles [11]. However, this integration also introduces security vulnerabilities, with Position Falsification Attacks emerging as a

significant concern [12]. These attacks involve the manipulation of broadcasted location information to create artificial traffic congestion or disrupt safety applications, posing a grave risk to road safety and transportation efficiency [13].

Traditional security mechanisms are often insufficient in detecting Position Falsification Attacks, as they primarily focus on external threats rather than insider attacks within VANETs. Therefore, there is a pressing need to develop advanced detection techniques capable of accurately identifying these attacks. Machine Learning (ML) presents a promising approach to address this challenge, leveraging the vast amount of data generated within VANETs to detect anomalous behaviours indicative of attacks [14]. By training ML models on labelled datasets or employing unsupervised learning techniques, researchers can extract meaningful patterns from the data and build robust detection systems capable of identifying position falsification attacks in real-time.

Furthermore, the integration of Explainable Artificial Intelligence (XAI) principles enhances the interpretability and trustworthiness of ML models [15]. XAI techniques enable stakeholders to understand the underlying decision-making process of ML models, providing insights into why certain predictions are made and facilitating validation by domain experts[16]. This transparency is crucial in the context of VANET security, where the consequences of false positives or negatives can have far-reaching implications on road safety and traffic management.

This research endeavours to enhance the interpretability of anomaly detection in Vehicular Ad-Hoc Networks (VANETs). While previous efforts have focused on achieving high detection rates, they often lacked insight into the decision-making process of the deployed models. Thus, this study aims to develop advanced detection systems capable of discerning anomalies within VANET while also offering transparency in the decision-making process.

1.3 Problem Statement

In the context of Vehicular Ad-Hoc Networks (VANETs), the detection and understanding of position falsification attacks pose significant challenges. While existing research has made strides in detecting such attacks using Machine Learning (ML) algorithms, there remains a critical gap in comprehending the decision-making process of these models. High detection rates alone do not provide insights into why certain decisions are made, limiting the ability to mitigate threats and ensure the reliability of VANETs effectively. Consequently, there is a pressing need to develop advanced detection systems that not only accurately identify anomalies within VANET data but also offer explainability in the decision-making process of the deployed models. Addressing this challenge requires a multifaceted approach that integrates ML techniques with Explainable Artificial Intelligence (XAI) principles to enhance the interpretability and transparency of detection systems. By bridging the gap between detection efficacy and interpretability, this research aims to pave the way for safer, more transparent, and more resilient VANETs, ultimately ensuring the security and well-being of all road users.

1.4 Solution Outline

The proposed solution to this problem is a generalized model to detect malicious nodes using Machine Learning and integrating XAI. this solution follows a multifaceted approach. First, the Dataset used in this research is the first public extensible dataset available in the field of VANET: VeReMi Dataset(Vehicular Reference Misbehavior Dataset)[17]. This generated dataset is processed and passed on to ML models, including anomaly detection algorithms and ensemble methods, to detect position falsification attacks. Additionally, we integrate XAI techniques, such as SHAP (SHapley Additive exPlanations) values and LIME (Local Interpretable Model-agnostic Explanations), to provide insights into the decision-making process of the ML models.

1.5 Thesis Organization

The remaining outline of this thesis is as follows: chapter 2 includes an overview of fundamental concepts of VANET and position falsification attack along with a literature review of related work in misbehaviour detection using machine learning approaches. Chapter 3 contains an outline of the proposed methodology and a brief discussion of the VeReMi dataset, followed by chapter 4, including experimental setup and discussion of results. In the end, chapter 5 gives a conclusion followed by possible future work on the proposed methodology.

CHAPTER 2

Background Review

2.1 Overview of VANET

The modern era has witnessed remarkable advancements in communication and technology, leading to the establishment of various networks. One such network is VANET, which holds immense potential in expanding road networks while ensuring driver's comfort, safety, and security. VANET offers numerous benefits, including enhancing road safety, reducing fuel consumption and CO2 emissions, alleviating traffic congestion, promoting eco-friendly driving practices, and providing convenience to drivers. Additionally, VANET facilitates commercial opportunities such as advertising nearby establishments and locating essential amenities like gas stations.

VANET operates with nodes that move freely within the network, resulting in dynamic changes in its topology as vehicles travel at high speeds. Each vehicle within the network operates independently and can communicate with any other node. VANETs can cover expansive geographical areas and are not constrained by limited battery storage or power supply. The network comprises both uniform and non-uniform regions: uniform regions occur when vehicles share similar speeds, paths, and directions for an extended period, typically observed on highways. Conversely, non-uniform regions encompass streets where vehicles have varied paths, directions, and speeds, interacting with multiple vehicles during their journey.

2.1.1 Types of Communication

Communication within VANET is classified as having brief, fleeting interactions with minimal delay. Vehicles within the network are outfitted with On-board Units (OBUs) to connect with Road-side Units (RSUs)[18]. The OBU utilizes Global Positioning System (GPS) technology to relay the vehicle's current position in the network. RSUs serve as the network's backbone, facilitating communication among vehicles.

VANET communication encompasses five distinct types:

1. Vehicle-to-Vehicle (V2V): Each vehicle can communicate with others by broadcasting messages to multiple nearby vehicles.

2. Vehicle-to-Infrastructure (V2I): Vehicles interact with nearby infrastructures like RSUs or central authorities to request services or update their status.

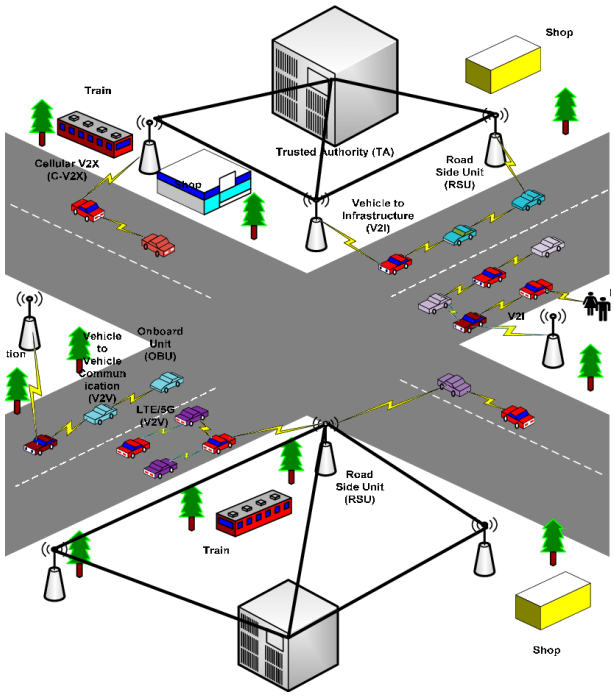

Figure 2.1: Types of communications in VANET [13]

3. Infrastructure-to-Infrastructure (I2I): Infrastructures communicate with one another to provide updated services to network nodes.

4. Infrastructure-to-Vehicle (I2V): Infrastructures communicate with vehicles to offer services, such as RSUs broadcasting hazard warnings to nearby vehicles.

5. Vehicle-to-Everything (V2X): Vehicles have the capability to communicate with various devices, including mobile phones and internet-connected devices.

The Figure 2.1 illustrates a smart transportation network integrating various communication technologies. Vehicles are equipped with Onboard Units (OBUs) for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication using LTE/5G and Cellular V2X (C-V2X) technologies. Roadside Units (RSUs) and a Trusted Authority (TA) manage and secure the communications. The setup aims to enhance traffic management, safety, and connectivity among vehicles, infrastructure, and pedestrians.

2.1.2 Security Requirements and Attacks in VANET

VANET provides numerous services through wireless channels; it also faces several drawbacks and is susceptible to security and privacy threats. Vulnerabilities within the network can lead to accidents and data loss, as wireless networks are susceptible to malicious attacks from individuals with different motives, as discussed in Section 1.1. This section examines the primary security requirements in VANET as outlined below [19]:

1. Authentication: Authentication verifies the legitimacy of members and their messages within the network[20]. Both senders and receivers must be authenticated members, and all information exchanged requires authentication to uphold network legitimacy. Examples of authentication attacks include Replay attacks, Certification Replication attacks, and Sybil attacks. In a Replay attack, an attacker sends the same message with a different time stamp [21]. In a Certification Replication attack, attackers possess replicas of vehicle keys to send false messages [13]. Sybil attacks involve attackers creating multiple identities or ghost vehicles to mislead legitimate vehicles with false messages [21].

2. Confidentiality: Confidentiality safeguards the information of registered nodes, ensuring that identities and geographical data remain protected. Only authenticated members should access network messages. Confidentiality attacks, like eavesdropping and information gathering, allow attackers to obtain private information for potential misuse. In an eavesdropping attack, the attacker silently listens to network communication and gathers data [22].

3. Integrity: Integrity ensures that information sent within the network remains unaltered before reaching the receiver. Message tampering by attackers is prohibited, including Message Deletion/Alteration and Timing attacks. In a Message Deletion/Alteration attack, attackers insert incorrect information or delete messages before reaching the receiver [12]. Timing attacks intentionally delay emergency messages [23].

4. Availability: Availability ensures uninterrupted services for legitimate nodes. Attacks on availability, like Denial of Service (DoS) attacks and jamming, disrupt network services. A spamming attack inundates the network with requests, rendering it unavailable [24]. DoS attacks make the network unavailable to authenticated members [25]. Jamming attacks disrupt the network by interfering with signals [26]. Broadcast tampering occurs when attackers insert erroneous messages, disturbing network functionality [19].

2.1.3 Position Falsification Attack

Vehicular Ad-Hoc Networks (VANETs) serve two primary applications: comfort and safety. Comfort applications offer services like weather updates, nearby gas stations, restaurants, and advertisements for user convenience. Safety applications focus on security, providing warnings like blind spots and hazard alerts. Vehicles broadcast their status via Basic Safety Messages (BSMs), digitally signed with the current position, speed, direction, and transmission time. BSMs are encrypted using cryptographic techniques, ensuring only authenticated members can decrypt them. However, malicious vehicles may send false position data in BSMs, potentially causing harmful

effects. These attacks, known as Position Falsification attacks[27], compromise data integrity by altering the vehicle's actual position. Such attacks can occur due to GPS errors or malicious intent from insiders or outsiders. Adequate security measures are crucial to mitigate the impact of these attacks and maintain network integrity.

1. Constant Position attack(CPA): In this attack, the sender vehicle continuously broadcasts fixed position coordinates in the BSM, pretending to be in the same network position. This attack could mislead honest vehicles into thinking of it as a hazard or traffic congestion on the road.

2. Constant Position offset Attack(CPOA): Attacker vehicle adds a constant offset/fixed value to the actual position and transmits the network's altered position. This attack is difficult to detect as the attacker is behaving normally by slightly altering the actual position in the BSM.

3. Random Position Attack(RPA): In a random position attack, the attacker sends a random position coordinate from the simulation area/playground in the network. It creates confusion in the network as every next BSM will have an entirely different and random value from the simulation.

4. Random Position Offset Attack(RPOA): Attackers send a random value from a preconfigured area around their vehicle. This attack is similar to a constant offset attack as both slightly alter the position information.

5. Eventual Stop Attack(ESA): The attacker tries to behave normally for some time in the network and then suddenly sends a fixed position repeatedly to depict an eventual stopping of the vehicle. Attackers mislead legitimate vehicles by gaining trust in the network for some time and then deceiving them.

2.2 Overview of Machine Learning

Machine learning, a branch of Artificial Intelligence, empowers machines to execute specific tasks efficiently by leveraging statistical learning[28]. Its applications span

various fields like healthcare, e-commerce, and law, facilitating disease detection, facial recognition, and email spam filtering. By identifying patterns in input data, machine learning algorithms make predictions, categorize information, and address real-world challenges[29]. Within Vehicular Ad Hoc Networks (VANETs), these algorithms are crucial in detecting attacks, intrusions, and network misbehaviour. Machine learning encompasses four primary types:

- Supervised Learning: This involves training algorithms with labelled data to address classification and regression problems.
- Unsupervised Learning: Algorithms work with unlabeled data, leveraging pattern recognition and similarity detection for tasks like clustering and anomaly detection.
- Semi-supervised Learning: Combining labelled and unlabeled data, this approach benefits from both supervised and unsupervised learning techniques.
- Reinforcement Learning[30]: Algorithms learn through interaction with their environment, receiving rewards for successful actions and learning from failures.

2.2.1 Basic Machine Learning Concepts and Terminologies

The basic terminologies and processes of machine learning[31] used in this thesis are defined below:

1. Feature: Features, also referred to as input variables, denote measurable properties or characteristics of the data under consideration. They serve as the basis for machine learning models to make predictions or classifications.
2. Label: In supervised learning scenarios, a label represents the output or target variable that the model seeks to predict based on the input features. Each data point in the training dataset is associated with a corresponding label.
3. Training Data: The training data set comprises a collection of input features along with their corresponding labels. This dataset is utilized to train machine

learning models, enabling them to learn patterns and relationships within the data.

4. Testing Data: Distinct from the training data, the testing dataset is employed to evaluate the performance and generalization capability of the trained model. Predictions generated by the model on the testing data are compared against actual labels to assess accuracy and efficacy.

5. Model: A model is a machine learning algorithm which is trained to solve a problem.

6. Testing/Evaluation: Evaluation involves assessing the performance of the trained model on unseen testing data. Various evaluation metrics, including accuracy, precision, recall, and F1-score, are utilized to gauge the model's efficacy.

2.2.2 Classification Algorithms

Classification, a subset of supervised learning, involves categorizing labelled input data into distinct classes. In the context of VANET, machine learning can be used to differentiate between legitimate vehicles and misbehaving nodes. The classification problem entails assigning data points to specific classes, with algorithms referred to as classifiers training models by identifying similarities within the dataset. This process aids in accurately categorizing the data for various applications.

- Binary classification: Binary classification involves the prediction of two distinct classes from a given data set. An example of this type of classification is spam detection. In this thesis, two classes in binary classification are legitimate vehicles and attacker vehicles.

- Multiclass classification: Multiclass classification involves classifying/predicting more than two classes in a dataset. Five different position falsification attacks and legitimate vehicles are the classes for multiclass classification in this research.

This section contains a brief description of the classification algorithms that were used in this research. We implemented K-nearest neighbour, Decision Tree, and Random Forest Algorithms.

2.2.2.1 K-Nearest Neighbours

K-Nearest Neighbour algorithm [32] is widely used for solving classification problems. It is suitable for balanced as well as imbalanced datasets. K-Nearest Neighbour works by finding the distance between all the points and a query point and selecting the nearest neighbours to a query point. Based on the labels of k nearest neighbours, it chooses the label based on popularity. This label is assigned to the query point by the majority vote of the neighbours. Distance between the points can be calculated using Euclidean, Manhattan, Minkowski or Hamming distance functions.

2.2.2.2 Decision Tree Algorithm

Decision Tree algorithm [33] constructs a tree of a dataset with branches to perform classification. The topmost node, known as the root node, corresponds to the best feature in the dataset. It consists of two entities, the decision node and the leaf node. A decision node is the condition on which a tree navigates, and leaf nodes are the outcomes of the decision node's conditions. The main advantage of this algorithm is it does not require any pre-processing of data and is faster. One major disadvantage is that it is more prone to overfitting.

2.2.2.3 Random Forest Algorithm

The Random Forest algorithm addresses classification and regression tasks as described in [34]. Comprising a collection of decision trees, this model predicts outcomes based on the dataset provided. By employing an ensemble method, the algorithm selects the best solution from the predictions generated by the individual trees. Random Forest mitigates the limitations of the Decision Tree algorithm and demonstrates robustness, yielding more accurate results in comparison.

2.3 Overview of XAI

Explainable Artificial Intelligence (XAI) represents a pivotal advancement in the field of machine learning, emphasizing the interpretability and transparency of AI models. As AI systems become increasingly complex, there is a growing need to understand the rationale behind their decisions, particularly in critical domains such as healthcare, finance, and autonomous systems. XAI addresses this imperative by providing human-understandable explanations for AI-driven predictions and recommendations, fostering trust, accountability, and regulatory compliance[35]. Techniques such as feature importance analysis, model-agnostic approaches like LIME and SHAP, rule-based systems, and interactive visualization tools enable users to comprehend and scrutinize AI models' decision-making processes. The different approaches in XAI are

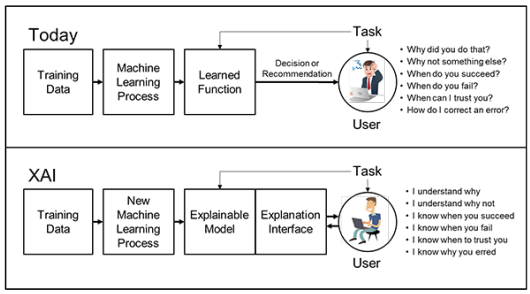

Figure 2.2: XAI Concept[36]

- Model-Agnostic Techniques: These techniques can be applied to any type of machine learning model. Approaches like feature importance analysis, identifying the most important features used by the model and LIME(Local Interpretable Model-agnostic Explanations) and SHAP(SHapley Additive Explanations) which provide the explanation of the model behaviour at a particular instance.

- Model-specific Techniques: These techniques are designed for specific types of machine learning models. For example, for decision trees, simply visualize the tree structure to understand how the model makes decisions.

This research focuses mainly on LIME and SHAP to provide explanations of the decision-making of the black-box models. The Figure 2.2 shows how a model can be more explainable to the users using the XAI model.

2.3.1 LIME

LIME is an XAI method that can explain the predictions of any classification or regression by approximating it locally with an interpretable model[16]. Usually, a black-box machine learning model takes data and produces outputs, but we can't see how it works. LIME tackles this by creating a simpler, transparent model around a specific data point. This local model tries to mimic the complex model's behaviour for that particular instance, revealing which features were most influential in the original model's prediction. LIME is model-agnostic, where it can be applied to any model like neural networks, decision trees, and support vector machines. LIME provides clear, understandable explanations of the model's decision-making.

2.3.2 SHAP

SHAP(SHapley Additive exPlanations) is another XAI technique like LIME. While LIME focuses on explaining specific instances, SHAP tries to explain both global model behaviour and local instance behaviour[37]. It used the game theory concept to distribute credit among all features for their contribution to the model's prediction. SHAP calculates how much each feature's presence impacts the prediction, providing a better understanding of the feature importance of the entire dataset. For local explanations, it considers all possible combinations of features and their respective contributions, averaging them to get SHAP values, which are used to provide explanations.

2.4 Basic Safety Messages

Basic Safety Messages (BSMs) play an important role in VANET safety applications[5]. They are commonly used in connected vehicle environments, where vehicles communicate with each other and roadside infrastructure to share information about their status and road conditions. BSM is a standardized communication packet sent every tenth of a second between connected vehicles via Dedicated Short Range Communication (DSRC)[6]. The data includes the sending vehicle's direction, speed, location, and turn signal status. The primary goal of BSM is to increase situational awareness. BSMs offer the potential to significantly improve road safety and reduce traffic congestion, ultimately paving the way for autonomous driving technologies.

BSMs typically contain the following crucial information:

- Vehicle ID: A unique identifier for the transmitting vehicle.
- Position: The vehicle's current location, often represented by latitude, longitude, and altitude.
- Speed: The vehicle's current speed.
- Heading: The vehicle's direction of travel.
- Acceleration: The vehicle's current acceleration or deceleration.
- Vehicle Status: Information about the vehicle's state, such as braking, turning, or lane changing.
- Timestamp: The time the BSM was generated.

2.5 VeReMi Dataset

The VeReMi dataset is a popular resource for researchers developing methods to detect misbehaving vehicles in VANETs (Vehicular Ad-Hoc Networks)[17]. This simulated dataset includes message logs from on-board units in vehicles. Each log con-

tains GPS data for the local vehicle and Basic Safety Messages (BSMs) received from other vehicles using DSRC (Dedicated Short-Range Communication)[27]. The dataset's strength lies in its realistic simulation of various traffic scenarios, encompassing both normal and malicious behavior. This comprehensiveness makes VeReMi widely adopted by the research community for studying misbehavior detection in VANETs.

2.5.1 Features of VeReMi

- **Extensible and Publicly Available**: The VeReMi dataset is the first public dataset of its kind that is extensible[27], allowing researchers to reproduce the data generation process and contribute additional attack scenarios. It also enables comparative analysis of new detection mechanisms against existing ones.

- **Detailed Simulation Executions**:The VeReMi dataset is generated using the LuST (Luxembourg traffic) scenario[38] and the VEINS[39]framework, which is built on OMNeT++[40] and SUMO. It comprises 225 simulation runs, categorized by vehicular density: low density includes 35 to 39 vehicles, medium density ranges from 97 to 108 vehicles, and high density has between 491 and 519 vehicles.

- **Reception Logs and Ground Truth Files:**Each vehicle in the VeReMi dataset has detailed reception logs and ground truth files, which capture essential data such as reception times, sender information, and position updates. These logs include both legitimate and malicious messages and are crucial for accurately evaluating misbehavior detection mechanisms.

2.6 Literature Review

Nowadays, many researchers are using a machine learning approach for misbehaviour detection or attack detection in VANET. Machine learning framework such as [41] focus on detecting the attacks but do not explain the decision-making process of the

machine learning model. Though machine learning models perform better than the traditional detection mechanism, every model cannot be trusted based on their prediction since they are black-box models. Additional techniques, such as explainable artificial intelligence, can help to interpret the black-box models. XAI helps to identify the insights of the model decision-making[35]. It helps in choosing the model with more trust. Some of the machine learning and explainable artificial intelligence approaches are discussed in this section. A comparative analysis of the literature review is addressed in Table 2.1.

2.6.1 Machine Learning in VANET Misbehavior Detection

Grover et al. [42] proposed an ensemble-based machine learning approach to detect misbehaviour in VANETs, combining the strengths of multiple classifiers to improve accuracy. The proposed WEKA framework classifies various types of misbehaviours using features extracted from nodes in NCTUns-5.0 simulator scenarios. The ensemble method outperforms individual base classifiers, as demonstrated. In [22], the authors examine the vulnerability of vehicular networks to attacks like DoS, Sybil, and false alerts. They also highlight the limitations of cryptographic methods in preventing insider attacks. The authors propose a system for detecting misbehaviour in vehicular networks using machine learning and simulation data. This approach outperforms previous methods in detecting various misbehaviours. The authors claim to achieve better accuracy with their method. According to the authors, Random Forest and Decision Trees outperformed other classifiers.

Khot et al. [43] proposed a machine-learning framework to predict the vehicle's next position in the network. The authors incorporated beacon messages from nearby vehicles to create features like the distance between sender and receiver. Machine learning algorithms were used to train and test the model. The authors compared predicted and actual values in the BSM and classified vehicles accordingly. If the position does not match the prediction, it is classified as an attacker vehicle. The authors claimed Random Forest outperforms other algorithms. The authors claimed that Random forest performs best among the other algorithms.

2.6.2 Detecting Position Falsification Attack

Xue et al.[44] proposed a trusted neighbour table to detect position spoofing attacks. The location verification scheme involves creating a TNT for each vehicle to record its neighbouring vehicle's updated location. To use TNT-based location verification, each node in the network must keep a TNT with the most recent location of its neighbours. The authors differentiated their TNT from the neighbour table by authenticating its contents. Nodes generate trust values in the table, with higher values indicating greater trustworthiness of neighbouring vehicles. The authors claim their approach is secure and efficient when there is no infrastructure involved.

The study[45] on Vehicular Ad Hoc Networks (VANETs) and Intelligent Transportation Systems (ITS) addresses privacy, network overhead, and security challenges. They investigate the effectiveness of trust models in detecting misbehaving nodes in VANETs. The research evaluates trust metric parameters using machine learning techniques. The receiver power coherency metric is highly effective at identifying nodes involved in fake position attacks. Simulation results show that the approach accurately differentiates between well-behaved and misbehaving individuals. Improved vehicle security in transportation systems.

Gyawali et al.[14] developed a misbehaviour detection model for false alert verification and position falsification attacks. This framework uses the sender-receiver pair approach. A false alert occurs when an attacker sends a false alert to nearby vehicles. Alerts include hazard condition notifications, vehicle stopping warnings, and emergency braking. The proposed framework includes a misbehaviour detection model for each vehicle. Each vehicle broadcasts detected results to its neighbours, which are combined to determine which vehicle should be removed from the network. The authors apply the Greenshield model[46], which uses a linear speed-density relationship to estimate continuous traffic. The receiver vehicle calculates changes in speed, position, distance, and RSSI value compared to the sending vehicle. The dataset includes all of these values as features, which are then analyzed using machine learning algorithms.

Mankodiya et al.[47] developed a misbehaviour detection model for position falsification attacks. This framework uses machine learning and the XAI approach. They have integrated XAI into the machine learning model for better interpretation of the model. The research employs various machine learning algorithms to detect five types of attacks: constant, constant offset, random, random offset, and eventual stop attacks. The results show high accuracies with random forest and decision tree algorithms. The study uses the VeReMi dataset for malicious detection in VANETs. The study [48]explored the integration of XAI techniques like LIME and SHAP for explanations of the model's decision-making. The study uses a dataset which is generated from Burst-ADMA.

Table 2.1: Comparison table of Misbehaviour detection in VANET's

No.	**Paper**	**Type of attack**	**VeReMi Dataset Used?**	**XAI used?**	**Approach**
1	Grover et al.[42]	false alert and Sybil attacks	No	No	Ensemble method
2	Khot et al.[43]	Postion falsification attack	Yes	No	Predicting new position
3	Sharma et al.[5]	Postion falsification attack	Yes	No	consecutive BSM approach
4	Steven et al.[49]	Position falsification attacks	Yes	No	Additional plausibility checks
5	Gyawali et al.[14]	Position falsification attacks	Yes	No	sender-receiver pair approach
6	Mankodiya et al.[47]	false alert and Position falsification attacks	Yes	Yes	XAI based approach
7	Idris et al.[48]	Speed falsification attacks	No	Yes	XAI techniques SHAP and LIME approach
8	**Proposed Method**	Postion falsification attacks	Yes	Yes	Custom Split approach and Integrating SHAP and LIME

The Table 2.1 is an overview of research papers, summarizing their approaches, findings and contributions in identifying position falsification attacks.

In this research, the proposed methodology uses the custom split approach and integrating SHAP and LIME for position falsification detection. Machine learning algorithms are used to classify legitimate vehicles and attacker vehicles, and XAI techniques provide LIME and SHAP explanations for better understanding of models decision-making and helps to trust the models prediction.

CHAPTER 3

Custom split BSM and XAI approach

3.1 Introduction

Misbehaviour detection is a method of identifying attacks on the VANET using various techniques. In this research, the proposed methodology aims to detect position falsification attacks on VANET using machine learning algorithms and integrating XAI techniques like LIME and SHAP for explainability. Vehicles transmit BSMs into the network, and all vehicles and infrastructures nearby receive these BSMs. BSMs contain information on vehicles' current status in the network. This information includes sender ID, position, speed, time, and a unique message ID. BSM can help to identify the behaviour of the vehicle in the network. For this proposed method collecting the BSMs data from the vehicles in the network is necessary. VANETs are vulnerable networks, so it is not practical to directly apply the proposed method in real time. So, the proposed method is implemented in a simulated environment using the VeReMi dataset, which consists of a collection of BSMs in the network.

In this thesis, the Proposed methodology aims to provide a comprehensive machine learning-based approach for identifying malicious BSMs within the vehicular network. A custom-split BSM approach is implemented to partition the dataset into unique senders for training and testing data to ensure model evaluation and performance. Explainable AI(XAI) techniques, such as Local Interpretable Model-agnostic

Explanations(LIME) and Shapley Additive Explanations(SHAP), are integrated to explain the decision-making of the black-box models. These XAI techniques provide transparent and interpretable insights into the model's predictions, enhancing the understanding and trust in the system's output. This approach aims to improve the security and reliability of vehicular networks by effectively identifying and mitigating malicious activities.

3.2 Proposed Architecture

Vehicles transmit BSMs periodically into the network. All the neighbouring infrastructure and vehicles can receive these BSMs. Cryptographic methods like encryption and decryption provide authentication of the BSM in the network. The Central Authority assigns public and private keys to vehicles upon registration. Registered automobiles utilize these keys to sign network messages with Digital Signature techniques. These solutions allow only authenticated cars in the network to transmit and receive BSMs.

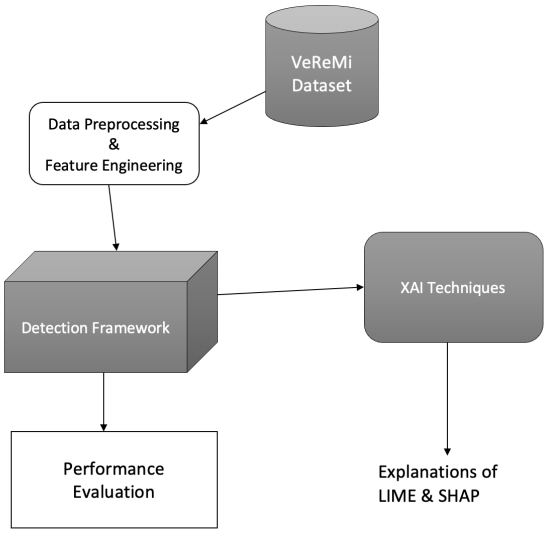

Figure 3.1: Proposed Architecture

The proposed architecture Figure 3.1 shows a model for detecting misbehaviour in vehicular networks and providing the explanations of decision-making of the black-

box models. The process begins with dataset preparation using the ground truth files and log files mapped to csv files, data preprocessing, and feature engineering, where vehicular data from the VeReMi dataset is cleaned, transformed and enhanced to create meaningful features. these features are then input into the detection framework, which employs machine learning algorithms to identify and classify the instances of the attacker among the vehicles. To provide transparency and explainability, XAI techniques such as LIME(Local Interpretable Model-agnostic Explanations) and SHAP(SHapley Additive exPlanations) are used. These techniques offer detailed explanations of the model's decision-making by highlighting the significance and impact of the individual features, making the detection process trustworthy.

3.3 High-level Outline of Proposed Approach

The proposed methodology has four main stages: dataset extraction, data preparation, classification and XAI Techniques, as shown in Figure 3.2. A detailed discussion of these three stages is as follows:

3.3.1 Data Extraction

VeReMi dataset has a total of 225 simulations with different traffic scenarios, and each simulation consists of two types of files they are ground truth file and log files. These simulations are particularly helpful for studying how safety and security systems perform in-vehicle networks. In a simulation of a vehicle's network activity, there is only one ground truth value. Ground truth includes an attacker type that categorizes genuine and misbehaving vehicles. Meanwhile, the number of logs in a simulation corresponds to the number of cars in the network.

Each vehicle generates a log containing all received BSMs from other cars. In a position falsification attack, attacker cars transmit false information in the BSM, resulting in false information in the log files. This setup allows researchers to study how misinformation can spread through the network, impacting communication and overall network safety. These log files are important for anyone looking to understand

and enhance the security measures in vehicular communications.

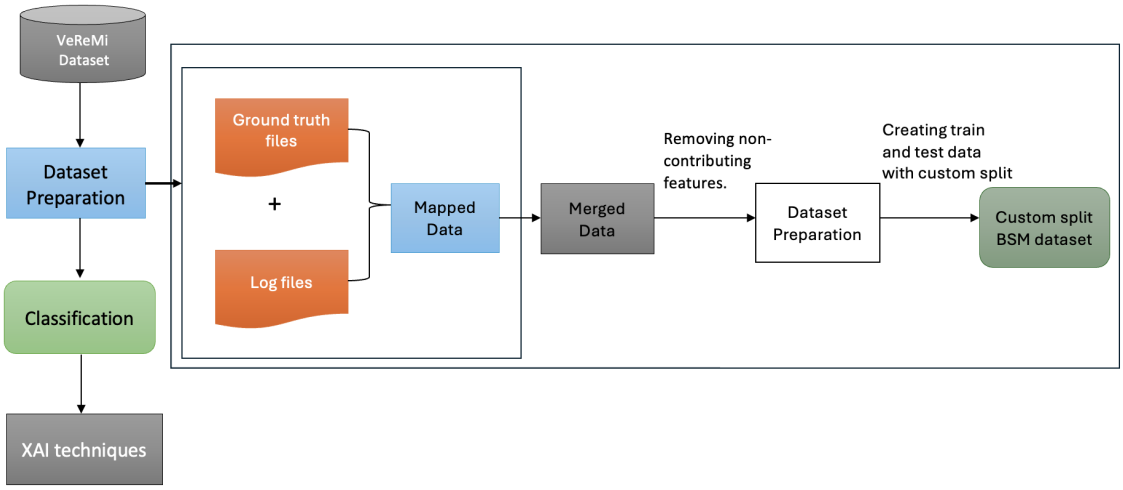

Figure 3.2: Proposed Methodology

To form a labelled dataset, The ground truth file is merged with log files for each simulation. During the data extraction, the ground truth file is mapped to the log file for each simulation. For a single simulation, the number of log files is equal to the number of receivers; hence, the first step is to combine these separate log files into a single file. The ground truth file and log files have a unique ID named messageID. To create a labelled dataset, the ground truth files attacker type must be mapped to data in the combined log file, as shown in Figure 3.3. There are five different seeds of the same scenario in the VeReMi dataset to create randomness in the network, and this process is repeated for five repetitions. All five repetitions were combined in the end to create a merged dataset for a single scenario.

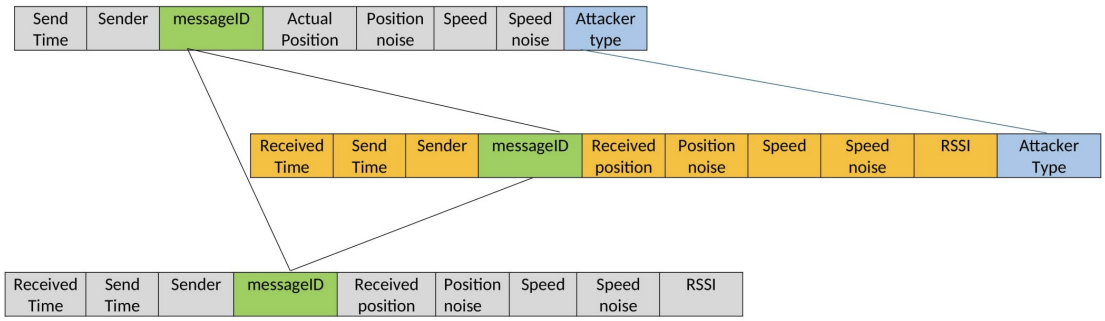

Figure 3.3: Data extraction of Ground truth file and Log files to create labelled data

3.3.2 Data Preparation

In this stage, merged data is pre-processed by removing any duplicate data present in the dataset and filtering non-contributing features. Non-contributing features are removed using the feature importance process. This process will provide information about the features which contribute more to detecting the behaviour of the vehicles and which contribute least. Non-contributing features can decrease the model's accuracy and efficiency, so it is best to remove those features.

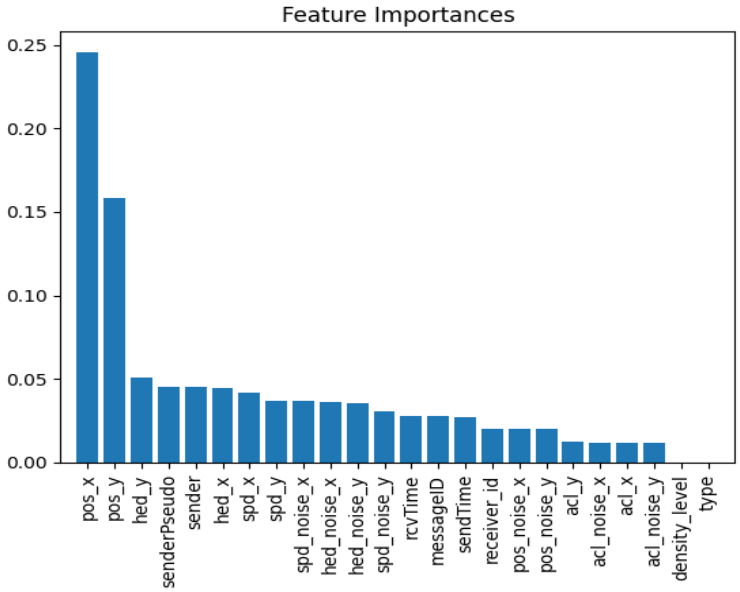

Figure 3.4: Feature importance

From the Figure 3.4, we can see that position, heading, and speed features contribute the most and other features have the least important information for training the model. The purpose of this approach is to find information on vehicles to detect misbehaviour in the network. We also removed a few features in the dataset that did not provide meaningful information before training the machine learning model. These features include position noise vectors, speed noise vectors, messageID, receiver-id, density-level, and Type.

Vehicles send BSMs into the network based on time, so the dataset contains multiple BSMs with the same vehicle according to the received time. so there is

a chance of data leakage possible with train and test data, and there is a chance for the algorithm to memorize the data points while training the model. So, a custom-split approach is used to separate the train and test data with a unique sender to avoid data leakage. This approach is done in two steps. Firstly, the data set is sorted by the receive time, and then unique senders are extracted and split into train and test data to make sure that there are unique senders in both train and test data. Implementing custom split can be useful to organize messages in chronological order to maintain the sequence of data and identify each vehicle by its unique ID to ensure separation between training and testing datasets and ensure that no vehicle data appears in both sets, preventing the model from memorizing specific patterns. Using custom split prevents overfitting by segregating senders, the model learns generalized attack patterns improving its ability to detect position falsification attack in unseen vehicles. It also provides realistic evaluation for real-world scenarios and efficiently handles different dataset sizes and network conditions, ensuring fairness and consistency.

3.3.3 Classification

The third stage of this methodology is to perform classification on the dataset. In this step, machine learning algorithms are implemented to classify the non-attacker vehicles from the network's attacker vehicles. In this thesis, binary classification will be performed on all five types of position falsification attacks. Machine learning algorithms like Random Forest, Decision Tree and K-Nearest Neighbour algorithms are used for classification. These algorithms train the model using a training set and classify the future data as non-attackers or attackers.

3.3.4 XAI techniques

In this stage, XAI techniques like LIME(Local Interpretable Model-agnostic Explanations) and SHAP(SHapley Additive exPlanations) are integrated to create explanations for the black-box model's decision-making. We first use LIME for local explanations since they are human-readable. LIME creates a local surrogate model

for each instance. surrogate models are simpler than the original model but try to mimic its prediction for the data being evaluated. LIME can reveal the original model's decision-making by assessing the surrogate model's prediction. such local explanations classify and explain each of the samples individually. We then provide human-understandable explanations for local samples. From Figure 3.5, we can see an example of a LIME explanation of a particular instance. This helps to understand how the model is trying to make the decision, whether it is an attacker or a non-attacker and also shows which feature contributes more to the decision-making of the model for a particular instance.

The LIME explanation in Figure 3.5 is for a constant position offset position falsification attack using random forest algorithm. Left side of the image indicates the confidence of the model for a attacker and non-attacker and the center tree like structure highlights the contributions of various feature to the prediction. orange colour indicates an attacker and blue indicates a non-attacker. The table on the right side has features and its value for this particular instance. For example in LIME explanation Figure 3.5 features pos-y(1138.18), pos-y(1429.75), senderspeudo(106935), acl-x(-1.03) and sendTime(50736.93) are top features influencing the BSM as attacker and features like hed-y(-0.86), messageID(1392944), rcvTime(50736.93), spd-y(-4,23) and hed-x(-0.52) are contributing to non-attacker with lesser extent. Overall, the LIME visualization provides a clear and interpretable explanation of the model's prediction, ensuring transparency and understanding of the machine learning model's decision-making process.

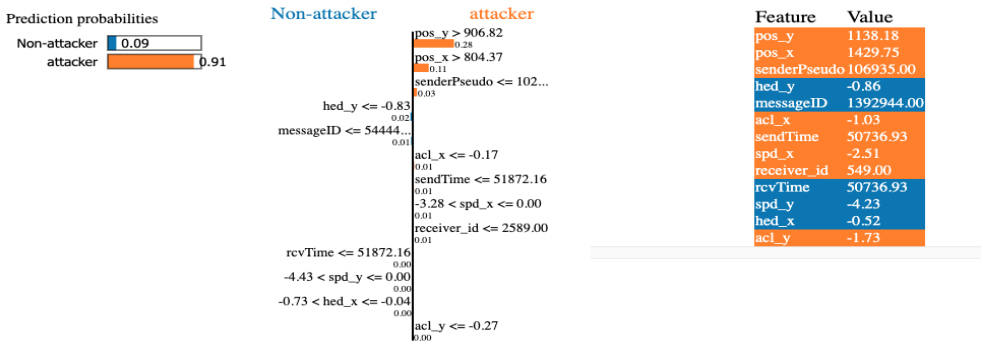

Figure 3.5: LIME Explanation

The next stage in our method is to provide global explanations. Figure 3.6 is a SHAP global explanation for a constant position offset using random forest algorithm. The SHAP summary plot provides a detailed interpretation of feature contributions to the machine learning model's predictions by leveraging a game theory to assign each feature an importance value called SHAP values. Unlike the traditional feature importance, which simply ranks features based on their contribution to the models accuracy, SHAP values offer more understanding by explaining how each feature affects the prediction. This plot highlights pos-x, pos-y as the most influential features, with SHAP values indicating that higher values in these features significantly impact the model's outputs.

SHAP values are calculated by considering the average marginal contribution of a feature across all possible feature combinations, ensuring fair allocation of importance. senderPseudo, along with heading(hed-y, hed-x) and speed(spd-y, spd-x), also demonstrates substantial impact. In contrast, features such as messageID, rcvTime, sendTime, receiver-id and acceleration(acl-x, acl-y) exhibit minimal influence, as their SHAP values remain closely clustered around zero, indicating negligible contribution to the model's output. This comprehensive analysis highlights the critical role of positional data and the moderate importance of sender identification and motion parameters, while timing, receiver and acceleration are less significant. These insights are important for enhancing model interpretability and guiding future model refinement efforts.

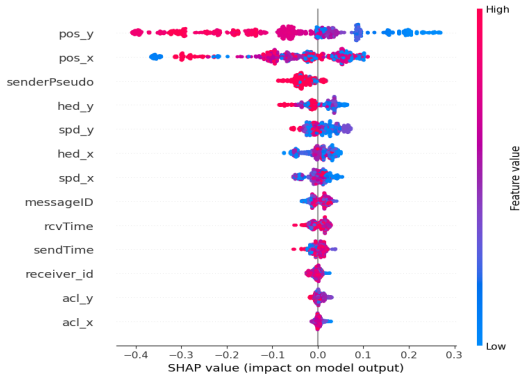

Figure 3.6: SHAP Explanation

3.4 How the Proposed Algorithm Differs from Existing Approaches

As discussed in the Section 2.6, many researchers have introduced a misbehaviour detection framework to detect position falsification attacks using the VeReMi dataset and Machine learning techniques. Techniques used in the current work include normalization of features, implementing plausibility checks and employing trust-based models to identify potential attacks. Some of the current work involves adding features of calculation, such as a change in speed and position to train the model, and most of the researchers split data randomly into train and test data, which may cause scenarios like overfitting and data leakage. The majority of the studies have worked on predicting the attacker and non-attacker vehicles in the network using machine learning algorithms.

Our work differs from the existing approaches in the following techniques:

- We sort BSMs by receive time and split the data based on unique vehicle senders. this ensures vehicles in the training set are not in the test set, preventing data leakage and mimicking real-world scenarios.
- This method forces the model to learn generalized attack patterns rather than memorizing specific vehicle data, enhancing its ability to detect falsification from unseen vehicles.
- We integrate LIME and SHAP for local and global explanations of the decision-making of the black-box models.

CHAPTER 4

Results

Due to safety concerns, high infrastructure costs, facilities, and resource requirements, conducting experiments to test the efficiency of a detection system in a real-world scenario is hazardous and difficult. As a result, we run such experiments on a digital scale using simulation tools. This is a much more cost-effective and safe way of evaluating and analyzing algorithms. In this chapter, the section 4.1 reviews the simulation setup of the VeReMi dataset, implementation environment and evaluation metrics. Sections 4.2, 4.3, and 4.4 discuss the results and explanations obtained, and section 4.5 discusses the comparison of the results with the existing approaches.

4.1 Setup Discussion

4.1.1 Simulation setup of VeReMi Dataset

In this research, we use the VeReMi dataset, which uses Luxembourg traffic scenario(LuST)[38] and offers a wide range of scenarios for evaluating the VANET application. The simulation parameters used to generate the VeReMi dataset are shown in the Table 4.1.

Table 4.1: Simulation parameters used in VeReMi dataset[27]

Parameters	Value	Description
Mobility	SUMO LuST	Luxembourg SUMO traffic
Simulation Area	2300, 5400-6300, 6300	Various road types
Simulation duration	100s	
Attacker probability	(0.1,0.2,0.3)	Attacker probability in the network
Simulation start	(3,5,7)h	Control density
Signal interference model	Two-ray interference	VEINS default
Obstacle shadowing	Simple	VEINS default
Shadowing	Log-normal	VEINS default
MAC implementation	802.11p	VEINS default
Thermal power	-110 dBm	VEINS default
Bit-rate	6 Mbps	VEINS default
Sensitivity	-89 dBm	VEINS default
Antenna model	Monopole on roof	VEINS default
Beaconing rate	1 Hz	VEINS default

4.1.2 Evaluation Metrics

To evaluate the performance of our proposed model, we use several key metrics, and they are discussed below.

- **True Positive(TP)** = instances correctly identified as positive(attacker)
- **True Negative(TN)** = instances correctly identified as negative(non-attacker)
- **False Positive(FP)** = instances incorrectly identified as positive(attacker)
- **False Negative(FN)** = instances incorrectly identified as negative(non-attacker)

Accuracy: accuracy is a metric which is used to evaluate the performance of a model. It is defined as the ratio of the correctly predicted observations to the total observations.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision: Precision measures the proportion of positive classifications that are actually correct. Precision is also called positive predicted value.

$$\text{Precision} = \frac{TP}{TP+FP}$$

Recall: Recall measures the ratio of actually positive classifications which was classified as positive. Recall is also known as sensitivity.

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1-score: F1-score is a harmonic mean of precision and recall. F1-score gives a trade-off between precision and recall such that a high F1-score denotes high precision and recall values.

$$\text{F1-score} = 2 * \frac{Precision*Recall}{Precision+Recall}$$

4.1.3 Implementation Environment and Toolkit

All the experiments in this research were conducted in the following environment and configuration:

- **Operatin system :** MacBook Pro - macOS Sonoma
- **Processor:** 3.2 GHz Apple M1
- **Memory:** 8 GB

Tools and libraries used for the implementation of this research are :

- **Programing language:** Python
- ‘**Integrated Development Environment:** Google Colab

- **Libraries:** Scikit-learn, matplotlib, NumPy, Pandas, XAI

4.2 Classification Results

We implemented three algorithms (K-Nearrest Neighbour, Random Forest, Decision Tree) and XAI techniques(LIME, SHAP) in the proposed framework on each attack type. In this section, we will discuss the classification results of three algorithms with precision, recall and F1-score as evaluation metrics for five types of position falsification attacks.

The Table 4.2 shows the classification results of the attack types of the position falsification attack. For **Constant Position Attack**, both the random forest and decision tree algorithm are performing better with 90 and 92 percent accuracies compared to the KNN algorithm with 86 percent.

The classification results of **Constant Position Offest Attack** show that all algorithms are performing well. Random forest and decision tree algorithms are 92 and 96 percent, slightly better than the constant position attack. KNN has an accuracy of 82 percent.

For **Random Position Attack**, the algorithms perform similarly to constant position attacks with accuracies of random forest and decision tree algorithms around 92 percent, and KNN performed better than constant position and constant position offset attacks with 89 percent accuracy.

For **Random Position Offset Attack**, the algorithms under-perform compared to other attacks. Random forest and decision tree algorithms have an accuracy of around 82 percent. and KNN has an accuracy of 62 percent, which is the lowest compared to other attacks.

For **Eventual Stop Attack**, the algorithms perform similarly to other attacks except random position offset attacks. Random forest and decision tree algorithms have accuracies like 92 and 90 percent, and KNN has an accuracy of 82 percent.

Table 4.2: Classification results of Proposed model

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Constant Position Attack				
Random Forest	92	92	91.25	89.71
Decision Tree	92.15	91.25	91.75	90.5
KNN	86.05	87.2	86.05	85.2
Constant Position Offset Attack				
Random Forest	92	92	92.25	92
Decision Tree	96	96	96	96
KNN	82	82	81	82
Random Position Attack				
Random Forest	92.5	92.5	92.25	92.5
Decision Tree	92.7	92.4	91.75	91.9
KNN	89.05	89.2	89.05	89
Random Position Offset Attack				
Random Forest	82	82.6	82.25	81.71
Decision Tree	83.2	83.2	82.2	82.5
KNN	63.05	62.2	62.05	62.2
Eventual Stop Attack				
Random Forest	92.5	92.6	92.5	92.4
Decision Tree	89.9	90	89.9	89.9
KNN	82.05	81.4	82.05	81.5

In the following sections, we use LIME and SHAP values to understand the decision-making process for the above classification results. Although we have used XAI for all three models, in the remainder of this chapter, we will focus on the explanations using the *random forest* model only. The results for the other 2 models can

be explained in a similar manner.

4.3 LIME EXPLANATIONS

LIME is used to explain individual predictions. In this section, we discuss some of the LIME explanations for the individual instances of position falsification attacks. We consider 5 different examples, one for each different type of position falsification attack discussed in Sec 2.1.3.

Figure 4.1: LIME explanation of Random forest CPA

The LIME explanation in Figure 4.1 provides an explanation that Random Forest predicts this instance as a non-attacker with 94 percent high confidence for *constant position* attack. This instance is a true negative case. The model has correctly classified the BSM. Top Features like pos-x(905.86) contribute most towards the attacker, while pos-y(526.59), spd-y(-10.29), sender-pseudo(102015.00), rcvTime(50548.44) contribute this as a non-attacker.

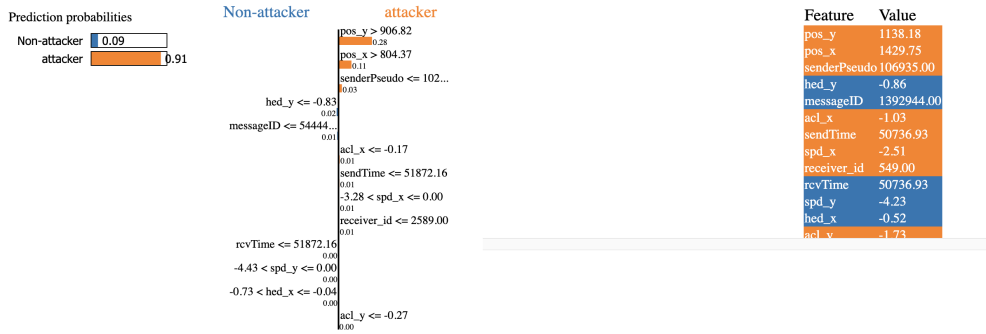

Figure 4.2: LIME Explanation of Random forest CPOA

LIME explanation in Figure 4.2 provides an explanation that random forest predicts this instance as an attacker with 91 percent high confidence for *constant position offset* attack. This instance is a true positive case, and the model correctly classifies the BSM. Features like pos-y(1138.18), pos-x(1429.75), senderPseudo(106935), acl-x(-1.03), and sendTime(50736.93) are the top features contributing to this BSM as an attacker.

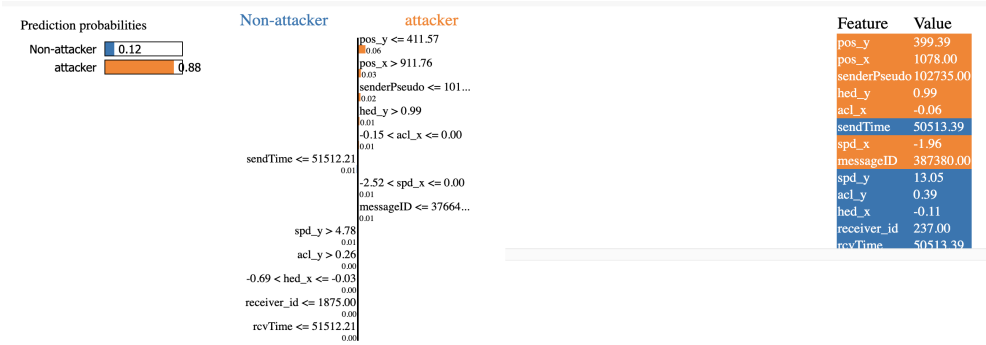

Figure 4.3: LIME Explanation of Random forest RPA

The LIME explanation in Figure 4.3 provides an explanation that the random forest model predicts this instance as an attacker with 88 percent confidence for *random position* attack. This instance represents a true positive case, where the model correctly classified the BSM. Top features contributing to this classification include pos-y (399.39), pos-x (1078.00), senderPseudo (102735), hed-y (0.99), and acl-x (-0.06).

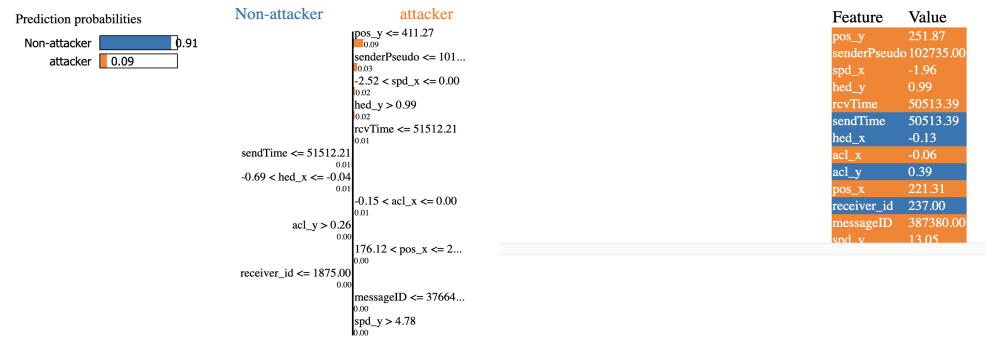

Figure 4.4: LIME Explanation of Random forest RPOA

The LIME explanation in Figure 4.4 provides an explanation that the random forest model predicts this instance as a non-attacker with 91 percent high confidence

for a *random position offset* attack. This instance represents a false negative case, Where the model incorrectly classified the BSM. Top features like pos-y(411.27), senderPseudo(102735.00), spd-x(-1.96), hed-y(0.99), rcvTime(50513.39) contribute as an attacker, but this model classified this instance as a non-attacker.

Figure 4.5: LIME Explanation of Random forest ESA

The LIME explanation in Figure 4.5 provides an explanation that the random forest model predicts this instance as an attacker with 95 percent high confidence for *eventual stop attack*. This instance represents a true positive case, where the model correctly classified the BSM. Top features like rcvTime(50513.39), sendTime(50513.39), messageID(387380), pos-y(136.75) contributes it as a attacker and other features like spd-y(0.00), pos-x(251.88), hed-x(-0.14), hed-y(0.99) contributes little towards non-attacker.

Throughout our analysis, we have utilized LIME explanations to find the behaviour of random forest algorithm in identifying various position falsification attacks, including CPA (Coordinate Position Attack), CPOA (Coordinate Position Offset Attack), RPA (Random Position Attack) and RPOA (Random Position Offset Attack). The LIME explanations reveal that features such as pos-x, pos-y, senderPseudo, rcvTime, sendTime, heading (hed), speed (spd), and receiverId are the most influential in these attack classifications. Specifically, pos-x and pos-y provide critical spatial information, senderPseudo identifies the unique sender, while rcvTime and sendTime offer temporal context. Additionally, heading and speed contribute to understanding the motion dynamics of the vehicle, and receiverId helps distinguish different receivers involved. For the ESA (Eventual Stop Attack) type, features like rcvTime,

sendTime, messageID, position coordinates (pos-x and pos-y), and speed are found to be the most impactful. These features are crucial for capturing the temporal sequence, unique message identification, spatial positioning, and velocity of the vehicle, which are essential for accurately detecting ESA attacks. By using these LIME explanations, we get a deeper understanding of the feature's importance and model behaviour, enhancing the transparency and interpretability of our attack detection system.

4.3.1 LIME explanations for TP, TN, FP and FN predictions

In this section, we are using LIME to analyze the performance of a random forest model in detecting constant position offset attacks. By examining True-Positive, True-Negative, False-positive and False-Negative classifications, LIME helps identify the key features influencing the model's decisions. This analysis provides insights into the model's strengths and weaknesses, guiding improvements for more accurate and reliable attack detection.

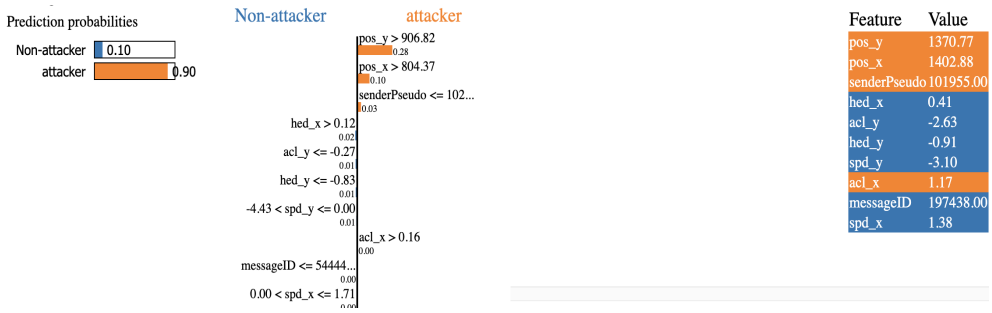

Figure 4.6: LIME Explanation of True-Positive instance of Random Forest

Figure 4.7: LIME Explanation of True-Negative of Random Forest

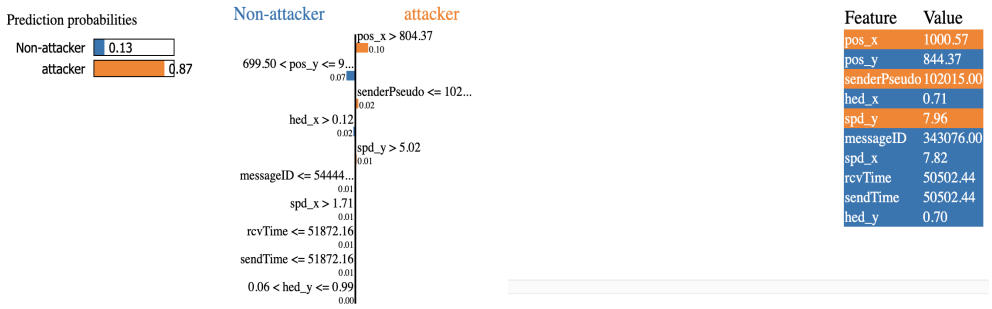

Figure 4.8: LIME Explanation of False-Positive instance of Random Forest

Figure 4.9: LIME Explanation of False-Negative instance of Random Forest

The Figures 4.6, 4.7, 4.8, 4.9 are the explanations of the True-positive, True-negative, False-positive, False-negative by random forest for the constant position offset attack. These explanations can help to get insights into models' decision-making in true and false cases. For example, Figure 4.9 shows the model predicted vehicle as a non-attacker with 77 percent confidence, despite most of the features like pos-y(916.13), pos-x(925.66), senderPseudo(101955), spd-x(-8.48) influence the vehicle as an attacker. This is a false negative case where the model has predicted the output incorrectly.

The Random Forest model occasionally misclassified certain instances due to the ensemble voting mechanism, where predictions are derived from the majority vote of multiple decision trees. For this instance, even though features like pos-y and pos-x indicate an attacker, their influence is counteracted by other features like spd-y, hed-y, acl-x, and acl-y, which align with non-attacker patterns. This happens because each tree in the forest may use different feature subsets and thresholds, leading to a majority vote that can favour non-attacker classification despite strong

individual indicators of an attack. Such mis-classification highlights the complex interplay of feature contributions and the variability in how trees within the random forest interpret feature values.

Table 4.3: Comparison of TP, TN, FP, and FN cases

Case Type	**Top Features**	**Feature Impact**	**Description**
True-Positive 4.6	pos-x, pos-y, sender, acl-x,spd,hed-x, hed-y,acl-y	Features correctly indicate an attacker	Model correctly classifies attack based on the top features indicating malicious behaviour with 90 percent confidence.
True-Negative 4.7	pos-y, pos-x, hed-x, sender, hed-y, messageID, acl-y, spd-x	pos-y, pos-x, sender misleadingly suggest attack; other features suggest non-attack	Despite some features suggesting an attack, the model correctly predicts as non-attack with 73 percent confidence.
False-Positive 4.8	pos-x, pos-y, hed-x, messageID, spd-x,rcvTime, hed-y, sender, spd-y	pos-x, sender, spd-x are misleading as an attacker and remaining features suggesting non-attacker	Model incorrectly predicts an attacker with 87 percent confidence despite most of the features suggesting a non-attacker
False-Negative 4.9	pos-y, pos-x, sender, spd-x, spd-y, hed-x, send time, acl-x, acl-y	pos-y, pos-x, sender, spd-x, hed-x, sendTime suggest as attacker and remaining as non-attacker	Model incorrectly predicts as non-attacker with 77 percent confidence although features suggest as an attacker.

The Table 4.3 is a summary of the comparison of the different prediction cases by random forest for constant position offset attack. Each row includes the case type, top features, features impact, and a brief description of each case.

Overall in this section, we examined true-positive, true-negative, false-positive, and false-negative instances using random forest. Random Forest provides detailed explanations and confidence levels. For instance, it correctly identified a true-positive attacker with 87 percent confidence influenced by features like pos-x, pos-y, acl-x, and spd-x. It also identified true-negative with 73 percent confidence despite most

of the features suggesting an attack. However, it misclassified a non-attacker as an attacker(false-positive) with 87 percent confidence and also misclassified a false-negative case as a non-attacker with 77 percent confidence. The random forest's detailed insights into these predictions highlight its effectiveness in understanding and improving model performance.

4.4 SHAP Explanations

These are some of the SHAP global explanations of the position falsification attack.

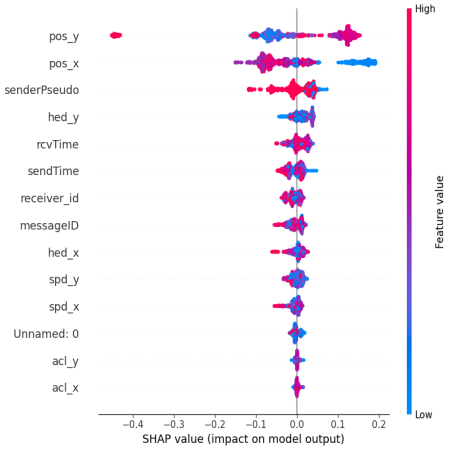

Figure 4.10: SHAP explanation of RF for CPA

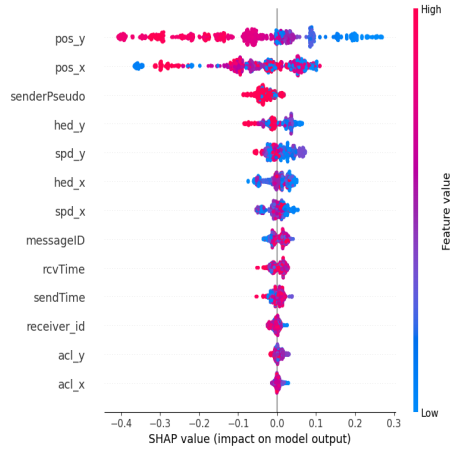

Figure 4.11: SHAP explanation of RF for CPOA

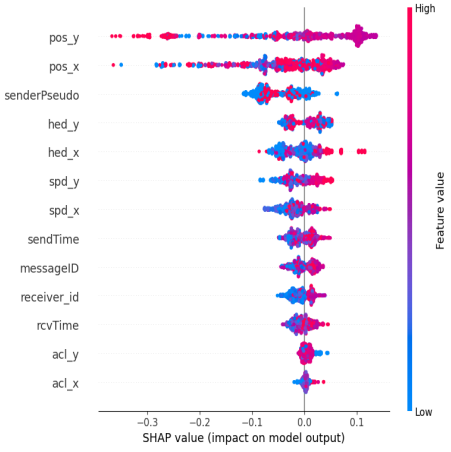

Figure 4.12: SHAP explanation of RF for RPA

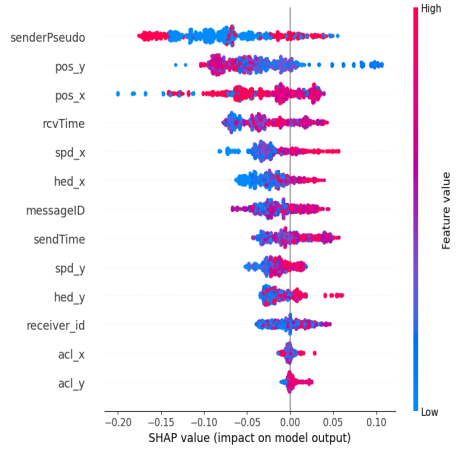

Figure 4.13: SHAP explanation of RF for RPOA

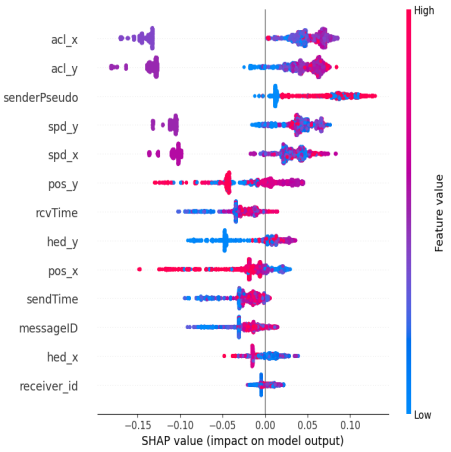

Figure 4.14: SHAP explanation of RF for ESA

The Figures 4.10, 4.11, 4.12, 4.13, 4.14 are the global explanations for five different attacks in position falsification attack using random forest algorithm. These SHAP summary plots explain the impact of each feature on the model's prediction by displaying how SHAP values for specific features influence the model's output. For instance, in Figure 4.11, which explains a constant position offset attack, the features pos-x and pos-y have the most impact on the model's output with a wide range of SHAP values. While other features like heading(hed-x, hed-y), senderPseudo, and speed(spd-x, spd–y) have a moderate impact on the model's output with less range of SHAP values. Features messageID, rcvTime, sendTime, receiver-id, acl-x, and acl-y have the lowest impact on the model's output, with SHAP values of almost zero. In this plot, red indicates the higher feature value, and blue indicates the lower feature value. This analysis highlights the critical role of positional data and the moderate importance of sender information and motion parameters, while timing, receiver and acceleration are less important. This detailed analysis is crucial for interpreting the model's behaviour and improving its robustness against position falsification attacks.

4.5 Comparison with Existing Approaches

Based on the performance of the different Machine learning algorithms and XAI explanations, we selected Random Forest with the proposed custom-split BSM and

XAI model to compare with the existing techniques. The proposed model is different from the existing models. We have used XAI techniques like LIME and SHAP to provide insights into the model's decision-making. Most of the existing approaches only focus on the classification of vehicles. We have tried to get insights into how the classification works because most ML models are black-box models which only provide the result but never tell how they got the result. A detailed review of these existing approaches can be found in the literature review Section 2.6

Table 4.4: Comparison of proposed model with existing approaches

Results from:	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Dataset**	**XAI Explanations**
Paper 1:[47]	98	99	97	98	VeReMi	Not provided
Paper 2: [48]	99	99	99	99	BurST adma	LIME and SHAP explanations provided
Proposed Model	92	92	92	92	VeReMi	LIME and SHAP explanations provided

Table 4.4 shows a comparison of the proposed approach with the existing techniques. Paper 1 and Paper 2 perform similarly with almost 98 and 99 percent accuracy. Paper 1 uses an XAI trust-based approach but does not provide any explanations for the model's predictions. This paper use VeReMi Datset and work on position falsification attacks. Paper 2 uses an ML algorithm and integrates XAI techniques like LIME and SHAP for providing explanations but uses a different dataset, which is derived from BurST-adma, and they have been implemented for speed falsification attacks. To the best of our knowledge, there are currently no XAI models for position falsification attacks for the VeReMi dataset.

The proposed model works on position falsification attacks using the VeReMi dataset, and we have implemented a new technique, a custom split BSM approach. We made sure the test data and training data were unique to avoid data leakage. The proposed model provides good classification results and LIME and SHAP explanations for the insights for a better understanding of the black-box ML models.

CHAPTER 5

Conclusion and Future Work

5.1 Conclusion

This thesis proposes a novel Machine learning and XAI techniques-based approach for classifying position falsification attacks and providing classification LIME and SHAP explanations in VANETs. Unlike the existing approaches, we have implemented a custom-split BSM approach to avoid data leakage. These custom-splitted BSM train data are used to train the proposed model using different machine learning algorithms. The test data is used to measure the performance metrics of the machine learning algorithms, and XAI techniques like LIME and SHAP are integrated to provide explanations of the classification made by the ML algorithms. The performance of the four different machine learning algorithms was compared with each other using performance metrics and explainability of the XAI explanations. Among all the machine learning algorithms, Random Forest and Decision tree algorithms yield the best results with an accuracy around 92 percent compared to the K-Nearest Neighbour algorithm. Random forest algorithm provides detailed LIME and SHAP explanations like the confidence of the model and important features that contribute to the model output compared to the decision tree and KNN algorithms. This thesis contributes by providing valuable insights into the model's decision-making by highlighting the significance and impact of individual features, making the detection process trustworthy. This work paves the way for future advancements in securing vehicular networks against cyber threats.

5.2 Future Work

The VeReMi dataset [17] is limited to five types of position falsification attacks and does not fully represent all possible attacks in VANETs. This approach can be implemented for all different types of attacks in the VeReMiEXT dataset [27] to provide useful insights. Feature engineering and hyper-parameter tuning can be applied to enhance the accuracy. Future work can be focused on enhancing the model's robustness against evolving attacks in VANETs, exploring advanced XAI techniques such as Gradients, DeepLIFT or model-specific interpretability methods for deeper insights. Develop methodologies for implementing real-time detection and responsive mechanisms based on insights from XAI explanations. Address privacy concerns related to collecting and using vehicular data in VANETs while ensuring the proposed approach remains scalable.

REFERENCES

[1] Sheng-hai An, Byung-Hyug Lee, and Dong-Ryeol Shin. "A survey of intelligent transportation systems". In: *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks*. IEEE. 2011, pp. 332–337.

[2] Social Determinants of Health. *Global status report on road safety 2018*. `https://www.who.int/publications/i/item/9789241565684`. [Online; accessed 17 June 2018]. 2018.

[3] Saif Al-Sultan et al. "A comprehensive survey on vehicular ad hoc network". In: *Journal of network and computer applications* 37 (2014), pp. 380–392.

[4] Sherali Zeadally et al. "Vehicular ad hoc networks (VANETS): status, results, and challenges". In: *Telecommunication Systems* 50.4 (2012), pp. 217–241.

[5] Aekta Sharma and Arunita Jaekel. "Machine Learning Based Misbehaviour Detection in VANET Using Consecutive BSM Approach". In: *IEEE Open Journal of Vehicular Technology* 3 (2022), pp. 1–14. DOI: `10.1109/OJVT.2021.3138354`.

[6] John B Kenney. "Dedicated short-range communications (DSRC) standards in the United States". In: *Proceedings of the IEEE* 99.7 (2011), pp. 1162–1182.

[7] Ghassan MT Abdalla, Mosa Ali Abu-Rgheff, and Sidi Mohammed Senouci. "Current trends in vehicular ad hoc networks". In: *Ubiquitous Computing and Communication Journal* (2007), pp. 1–9.

[8] Federico Poli. "Vehicular communications: from DSRC to Cellular V2X". PhD thesis. Politecnico di Torino, 2018.

[9] Parul Tyagi and Deepak Dembla. “A taxonomy of security attacks and issues in vehicular ad-hoc networks (vanets)”. In: *International Journal of Computer Applications* 91.7 (2014).

[10] Sunilkumar S Manvi and Shrikant Tangade. “A survey on authentication schemes in VANETs for secured communication”. In: *Vehicular Communications* 9 (2017), pp. 19–30.

[11] Maxim Raya and Jean-Pierre Hubaux. “Securing vehicular ad hoc networks”. In: *Journal of computer security* 15.1 (2007), pp. 39–68.

[12] Irshad Ahmed Sumra, Iftikhar Ahmad, Halabi Hasbullah, et al. “Classes of attacks in VANET”. In: *2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC)*. IEEE. 2011, pp. 1–5.

[13] Muhammad Sameer Sheikh and Jun Liang. “A comprehensive survey on VANET security services in traffic management system”. In: *Wireless Communications and Mobile Computing* 2019 (2019).

[14] Sohan Gyawali and Yi Qian. “Misbehavior detection using machine learning in vehicular communication networks”. In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[15] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778.

[16] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. ““Why Should I Trust You?”: Explaining the Predictions of Any Classifier”. In: Aug. 2016, pp. 1135–1144. DOI: 10.1145/2939672.2939778.

[17] *VeReMi dataset — VeReMi-dataset.github.io*.

[18] Gagan Deep Singh et al. "A Review on VANET Routing Protocols and Wireless Standards". In: *Smart Computing and Informatics*. Ed. by Suresh Chandra Satapathy, Vikrant Bhateja, and Swagatam Das. Singapore: Springer Singapore, 2018, pp. 329–340. ISBN: 978-981-10-5547-8.

[19] Mohammed Ali Hezam et al. "Classification of security attacks in VANET: A review of requirements and perspectives". In: (2018).

[20] Ram Shringar Raw, Manish Kumar, and Nanhay Singh. "Security challenges, issues and their solutions for VANET". In: *International journal of network security & its applications* 5.5 (2013), p. 95.

[21] Deepak Kushwaha, Piyush Kumar Shukla, and Raju Baraskar. "A survey on Sybil attack in vehicular ad-hoc network". In: *International Journal of Computer Applications* 98.15 (2014).

[22] Ajay N Upadhyaya and JS Shah. "Attacks on vanet security". In: *Int J Comp Eng Tech* 9.1 (2018), pp. 8–19.

[23] Irshad Ahmed Sumra, JAMALUL-LAIL Ab Manan, and Halabi Hasbullah. "Timing attack in vehicular network". In: *Proceedings of the 15th WSEAS International Conference on Computers, World Scientific and Engineering Academy and Society (WSEAS), Corfu Island, Greece*. 2011, pp. 151–155.

[24] Muhammad Rizwan Ghori et al. "Vehicular ad-hoc network (VANET)". In: *2018 IEEE international conference on innovative research and development (ICIRD)*. IEEE. 2018, pp. 1–6.

[25] Halabi Hasbullah, Irshad Ahmed Soomro, et al. "Denial of service (DOS) attack and its possible solutions in VANET". In: *International Journal of Electronics and Communication Engineering* 4.5 (2010), pp. 813–817.

[26] S Balasubramani, SK Rani, and K Suja Rajeswari. "Review on Security Attacks and Mechanism in VANET and MANET". In: *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 2016, pp. 655–666.

[27] Rens W van der Heijden, Thomas Lukaseder, and Frank Kargl. "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets". In: *International Conference on Security and Privacy in Communication Systems*. Springer. 2018, pp. 318–337.

[28] Mohssen Mohammed, Muhammad Badruddin Khan, and Eihab Bashier Mohammed Bashier. *Machine learning: algorithms and applications*. Crc Press, 2016.

[29] Vineet Chaoji, Rajeev Rastogi, and Gourav Roy. "Machine learning in the real world". In: *Proceedings of the VLDB Endowment* 9.13 (2016), pp. 1597–1600.

[30] Marco A Wiering and Martijn Van Otterlo. "Reinforcement learning". In: *Adaptation, learning, and optimization* 12.3 (2012), p. 729.

[31] Ethem Alpaydin. *Machine learning*. MIT press, 2021.

[32] Antonio Mucherino, Petraq J Papajorgji, and Panos M Pardalos. "K-nearest neighbor classification". In: *Data mining in agriculture*. Springer, 2009, pp. 83–106.

[33] Anuja Priyam et al. "Comparative analysis of decision tree classification algorithms". In: *International Journal of current engineering and technology* 3.2 (2013), pp. 334–337.

[34] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest". In: *R news* 2.3 (2002), pp. 18–22.

[35] G. Pradeep Reddy and Y. V. Pavan Kumar. "Explainable AI (XAI): Explained". In: *2023 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. 2023, pp. 1–6. DOI: 10.1109/eStream59056.2023.10134984.

[36] David Gunning and David Aha. "DARPA's explainable artificial intelligence (XAI) program". In: *AI magazine* 40.2 (2019), pp. 44–58.

[37] Scott M Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Advances in neural information processing systems* 30 (2017).

[38] Lara Codecá et al. "Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation". In: *IEEE Intelligent Transportation Systems Magazine* 9.2 (2017), pp. 52–63.

[39] Christoph Sommer et al. "Veins: The open source vehicular network simulation framework". In: *Recent Advances in Network Simulation*. Springer, 2019, pp. 215–252.

[40] Andras Varga. "OMNeT++". In: *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[41] Pranav Kumar Singh et al. "Machine learning based approach to detect position falsification attack in vanets". In: *International Conference on Security & Privacy*. Springer. 2019, pp. 166–178.

[42] Jyoti Grover, Vijay Laxmi, and Manoj Singh Gaur. "Misbehavior detection based on ensemble learning in vanet". In: *International Conference on Advanced Computing, Networking and Security*. Springer. 2011, pp. 602–611.

[43] Ankita Khot and Mayank Dave. "Position Falsification Misbehavior Detection in VANETs". In: *Mobile Radio Communications and 5G Networks*. Springer, 2020, pp. 487–499.

[44] Yiwen Ji. "A trusted neighbor table based location verification for VANET routing". English. In: *IET Conference Proceedings* (), 1–5(4). URL: https://digital-library.theiet.org/content/conferences/10.1049/cp.2010.0603.

[45] Muath Obaidat et al. "Security and Privacy Challenges in Vehicular Ad Hoc Networks". In: Jan. 2020, pp. 223–251. ISBN: 978-3-030-36166-2. DOI: 10.1007/978-3-030-36167-9_9.

[46] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Mosa Ali Abu-Rgheff. "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks". In: *IEEE Internet of things journal* 1.6 (2014), pp. 570–577.

[47] Harsh Mankodiya et al. "XAI-AV: Explainable Artificial Intelligence for Trust Management in Autonomous Vehicles". In: *2021 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. 2021, pp. 1–5. DOI: 10.1109/CCCI52664.2021.9583190.

[48] Hussaini Aliyu Idris et al. "Explaining Machine Learning Based Speed Anomaly Detection System Using eXplainable Artificial Intelligence". In: ().

[49] Steven So, Prinkle Sharma, and Jonathan Petit. "Integrating plausibility checks and machine learning for misbehavior detection in VANET". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2018, pp. 564–571.

VITA AUCTORIS

NAME:	Mahesh Abburi
PLACE OF BIRTH:	Chittoor, Andhra Pradesh, India
EDUCATION:	B.Tech in computer science, Anna University, Vellore, Tamil Nadu, India, (2021)
	M.Sc. Computer Science, University of Windsor, Windsor, Ontario, Canada, 2024