People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Béjaia
Faculty of Exact Sciences
Department of Computer Science

Master Thesis

In partial fulfillment of the requirements for the Master's degree in Computer Science

Specialization: Networks and Security

Theme

An AI-Based Intrusion Detection System for Identity-Based Advanced Persistent Threats in IoV Networks

Presented by:

Laribi Saïda
Habel Fairouz

Defended on Juin 30th before the jury:

President: Mrs. ALOUI Soraya
Examiners:
Mrs. MAAMERI Souhila
Mrs. YAICI Malika
Mrs. ZAMOUCHE Djamila

Supervisor: Pr. BOUALLOUCHE Louiza
Co-supervisor: Dr. OUYAHIA Samira

Academic Year: 2024–2025

Acknowledgment

Above all, we thank ***God Almighty*** *for granting us the strength, courage, and patience to carry out this work.*

We would like to express our deepest gratitude to our supervisors, ***Pr. BOUALLOUCHE Louiza*** *and* ***Dr. OUYAHIA Samira****, for their valuable guidance, unwavering support, and insightful advice throughout this research. Their availability, scientific rigor, and encouragement were instrumental in shaping the direction of this thesis and bringing it to completion.*

We also extend our sincere thanks to all the jury members for their time, attention, and constructive feedback. Their thoughtful observations and suggestions have not only helped improve the quality of this work but also deepened our understanding of the subject matter.

Finally, we are grateful to all the professors, administrative staff, and academic mentors who accompanied us throughout our academic journey. Their efforts and dedication provided us with the knowledge, tools, and environment necessary to grow and succeed.

Dedication

To my beloved parents

Thank you for your endless patience, your sacrifices, and your unwavering belief in me. Your constant support, whether emotional, moral, or financial, has been the foundation of this achievement. This work is as much yours as it is mine.

To my friends

For your help, motivation, and presence during the most stressful days, your support did not go unnoticed and will always be appreciated.

To my partner

For being a great support. Your dedication and reliability made this journey more meaningful.

Habel Fairouz

Dedication

To my dear parents

Thank you for everything. For your sacrifices, your unconditional love, your patience, and your constant support. None of this would have been possible without you.

To my siblings Mimi, Didi, Hlolo

Thank you for always being there with your encouragement, your help, and your quiet strength during the toughest moments.

To myself

For holding on, for not giving up, and for pushing through even when things felt overwhelming.

And to my partner

Your support, dedication and reliability made this journey more meaningful

To my friends

Thank you for your kindness, motivation, and for bringing light moments when I needed them most. Your presence made this experience less stressful and more joyful.

.

Laribi Saïda

Contents

List of Figures

List of Tables

List of acronyms

AI Artificial Intelligence

APT Advanced Persistent Threat

AUC Area Under the Curve

BCE Binary Cross-Entropy

BiGRU Bidirectional Gated Recurrent Unit

BiLSTM Bidirectional Long Short-Term Memory

BS Base Stations

BSM Basic Safety Message

CA Certification Authority

CAN Controller Area Network

CNN Convolutional Neural Network

DBSCAN Density-Based Spatial Clustering of Applications with Noise

DDoS Distributed Denial of Service

DL Deep Learning

DoS Denial of Service

DQN Deep Q-Networks

DSRC Dedicated Short-Range Communications

ECU Electronic Control Unit

EV Electric Vehicle

FN False Negative

FNN Feedforward Neural Network

FP False Positive

GAN Generative Adversarial Network

GAT Graph Attention Network

GPS Global Positioning System

GRU Gated Recurrent Unit

HU Head Unit

ID Identity

IDPS Intrusion Detection and Prevention Systems

IDS Intrusion Detection System

IoT Internet of Things

IoV Internet of Vehicles

IPsec Internet Protocol Security

ITS Intelligent Transportation System

LSTM Long Short-Term Memory

LTE Long-Term Evolution

MAC Message Authentication Code

MDS Misbehavior Detection System

ML Machine Learning

MLP Multilayer Perceptron

NN Neural Networks

NFC Near Field Communication

NS2 Network Simulator 2

OBU On-Board Unit

QoS Quality of Service

RFID Radio Frequency Identification

RL Reinforcment Learning

RNN Recurrent Neural Network

ROC Receiver Operating Characteristic

RSU Roadside Unit

SE Squeeze-and-Excitation

SUMO Simulation of Urban MObility

SVM Support Vector Machine

TA Trusted Authority

TCN Temporal Convolutional Network

TMC Traffic Management Center

TN True Negative

TP True Positive

V2I Vehicle-to-Infrastructure

V2P Vehicle-to-Pedestrian

V2R Vehicle-to-Roadside Unit

V2V Vehicle-to-Vehicle

V2X Vehicle-to-Everything

VeReMi Vehicular Misbehavior Detection

VANET Vehicular Ad Hoc Network

Wi-Fi Wireless Fidelity

WSN Wireless Sensor Network

XAI Explainable Artificial Intelligence

General Introduction

The rise of connected vehicles has introduced a new frontier in cybersecurity threats, where attackers increasingly exploit identity systems to launch stealthy, long-term intrusions. These threats are especially potent in the IoV, where communication protocols must constantly adapt to varying road contexts and application needs [1]. This high degree of adaptability makes IoV networks fertile ground for identity-based APTs, which leverage stolen or spoofed credentials to maintain persistent access while evading detection, sometimes for months before causing damage.

Current IDSs fail to catch these threats for several reasons. First, they often rely on static signatures that cannot adapt to the evolving topologies and dynamic conditions of vehicular networks [2]. Second, they focus predominantly on detecting network anomalies rather than behaviors that indicate misuse of legitimate identities. Third, they lack contextual awareness of vehicular movement patterns, such as direction, speed, and link stability, which have been shown to significantly affect routing decisions and communication reliability [3, 4].

This work addresses these limitations through three key innovations. First, we introduce behavioral fingerprinting, which detects APTs by analyzing subtle inconsistencies in vehicle motion and communication patterns—an approach grounded in the same principles used for assessing vehicular link quality and mobility-aware routing. Second, we implement a temporal-aware AI architecture that combines Temporal Convolutional Networks (TCNs) for local temporal patterns, Transformers for global contextual modeling, and Bidirectional GRUs (BiGRUs) for sequential analysis. Third, we emphasize real-world feasibility by using the VeReMi dataset's simulated but physically-grounded attack scenarios to bridge the gap between simulation and deployment.

This research matters because it prevents attackers from weaponizing vehicle identities to manipulate traffic flow or cause physical harm. It offers the first framework specifically designed to detect multi-stage identity attacks in IoV and achieves over 98% detection accuracy with a false negative rate under 1%, a critical threshold for safety systems.

The rest of this thesis is structured to first establish foundational knowledge in Chapter 1, then analyze existing solutions in Chapter 2, and finally present the proposed detection system in Chapter 3. Each section builds toward deployable solutions for automakers and smart city operators facing these emerging identity-based threats.

This work is distinct in several ways. It focuses exclusively on identity-based threats in IoV networks. It models APT behavioral patterns rather than relying solely on known attack signatures. It validates its effectiveness against attacks that typically bypass traditional vehicular IDS. Finally, the model is optimized for real-time use in resource-constrained in-vehicle

computing units, making it practical for on-road deployment.

In summary, this thesis provides security teams with their first specialized toolkit against one of the most dangerous and previously undetectable classes of vehicular cyber threats.

Chapter 1

Overview of Foundational Concepts

1.1 Introduction

IoV connects modern vehicles to smart infrastructure and networks, enabling real time access to information for both drivers and passengers. However, this high level of connectivity and the extreme reliance on data gathered from intelligent transportation systems introduce serious security vulnerabilities. IoV environments have become attractive targets for cyberattacks, some of which may take the form of APT-like behaviors. Stealthy, long term attacks where adversaries exploit identity related tactics such as impersonating legitimate vehicles or misusing credentials, allowing attackers to remain undetected by mimicking normal behavior. In this work, we focus on identity driven threats that exhibit APT like persistence, though their full lifecycle (e.g., reconnaissance, evasion) is inferred from behavioral anomalies in the absence of ground-truth identity labels.

Traditional IDS often fall short in detecting such threats, particularly in the dynamic and distributed nature of IoV. Their dependence on predefined signatures or basic anomaly models limits their ability to catch subtle, evolving attacks. To address this, recent research has focused on incorporating Deep Learning (DL) into IDS frameworks, especially temporal models like Recurrent Neural Network (RNN)s and Transformers, which can process large volumes of IoV data, recognize identity misuse patterns, and adapt in real time, providing a promising direction for countering identity-based deception.

This chapter lays the groundwork for the study. It begins by introducing the IoV and its core architecture, followed by an overview of APT-like threats and their operationalization in IoV through identity exploitation. The chapter concludes with a discussion on the limitations of traditional IDS and the specific advantages of DL for behavioral anomaly detection, setting the stage for Chapter 2's review of DL approaches and Chapter 3's hybrid model.

1.2 Internet of Vehicles (IoV)

1.2.1 IoV Definition

As an important branch of the Internet of Things (IoT) in the transportation field, IoV is a vast, dynamic mobile communication system consisting of numerous nodes connected to the network. These nodes include autonomous or human-driven intelligent vehicles, as well as other connected devices that exchange data through Vehicle-to-Everything (V2X) communications. IoV draws from multiple fields such as networking, communication, artificial intelligence, and cybersecurity to deliver intelligent applications. It enables gathering, sharing, transmitting, storing, computing, and analyzing data, which helps optimize traffic management and provide

intelligent mobile services in real time [5].

The IoV is considered a superset of Vehicular Ad Hoc Network (VANET)s, enhancing their capabilities by integrating advanced IoT connectivity. This integration supports more intelligent, scalable, and real-time communication between vehicles, infrastructure, and external networks. Both IoV and VANET share the goal of improving road safety and the overall driving experience [6, 7].

1.2.2 IoV Components

In IoV networks, various nodes communicate through essential electronic components that make up the system's architecture.

1. **Intelligent Vehicle** An intelligent vehicle is equipped with electronic components that manage and process data, enabling communication with other vehicles and infrastructure. These components facilitate message exchange and the recording of important traffic parameters like speed and driving behavior [8]. Key components include:

    - **Sensors**: gather data about the vehicle and its surroundings, such as temperature, tire pressure, and speed.
    - **Communication Systems**: connect vehicles to communication networks for data exchange with data centers, infrastructure, and other vehicles.
    - **Software**: analyzes gathered data to provide relevant information to drivers and transportation systems.
    - **Navigation Devices**: allow trip planning and provide directions.
    - **Safety Systems**: include autonomous braking, collision warning, and driver fatigue monitoring.
    - **Energy Management Systems**: control energy consumption, including battery recharge.
    - **User Interfaces**: Support human-machine interactions via voice commands, touchscreens, and other means [9].

2. **Road Components**

    - **Roadside Unit (RSU)**: Roadside devices that serve as access points for Vehicle-to-Infrastructure (V2I) communications.
    - **On-Board Unit (OBU)**: Electronic devices installed in vehicles, containing software for calculating and displaying necessary information.
    - **Base Stations (BS)**: Sites equipped with antennas that facilitate communication with mobile devices for network access [8].

1.2.3 IoV Architecture

Because there are various interpretations of IoV, no single, universally accepted architecture exists[10]. However, researchers in [11] propose a three-level architecture based on the interactions of different technologies within the IoV environment. Figure 1.1, adapted from [5], illustrates these layers:

1. **Perception Layer** This layer includes all sensors embedded in the vehicle, which collect environmental data and identify specific events such as driving behavior, vehicle status, and surrounding conditions, among others.

2. **Communication/Network Layer** The second layer supports various wireless communication modes, including Vehicle-to-Pedestrian (V2P), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Vehicle (V2V). It enables seamless and reliable connectivity to both existing and emerging networks like Wireless Fidelity (Wi-Fi) and Bluetooth.

3. **Application/Decision Layer** The top layer consists of data processing, storage infrastructure, and statistical tools that enable intelligent decision-making. It allows vehicles to access content and perform computations using Big Data technologies. This layer handles risk management related to hazardous road conditions and traffic congestion by integrating data from cloud computing, Wireless Sensor Network (WSN), and other technologies to produce unified decisions.

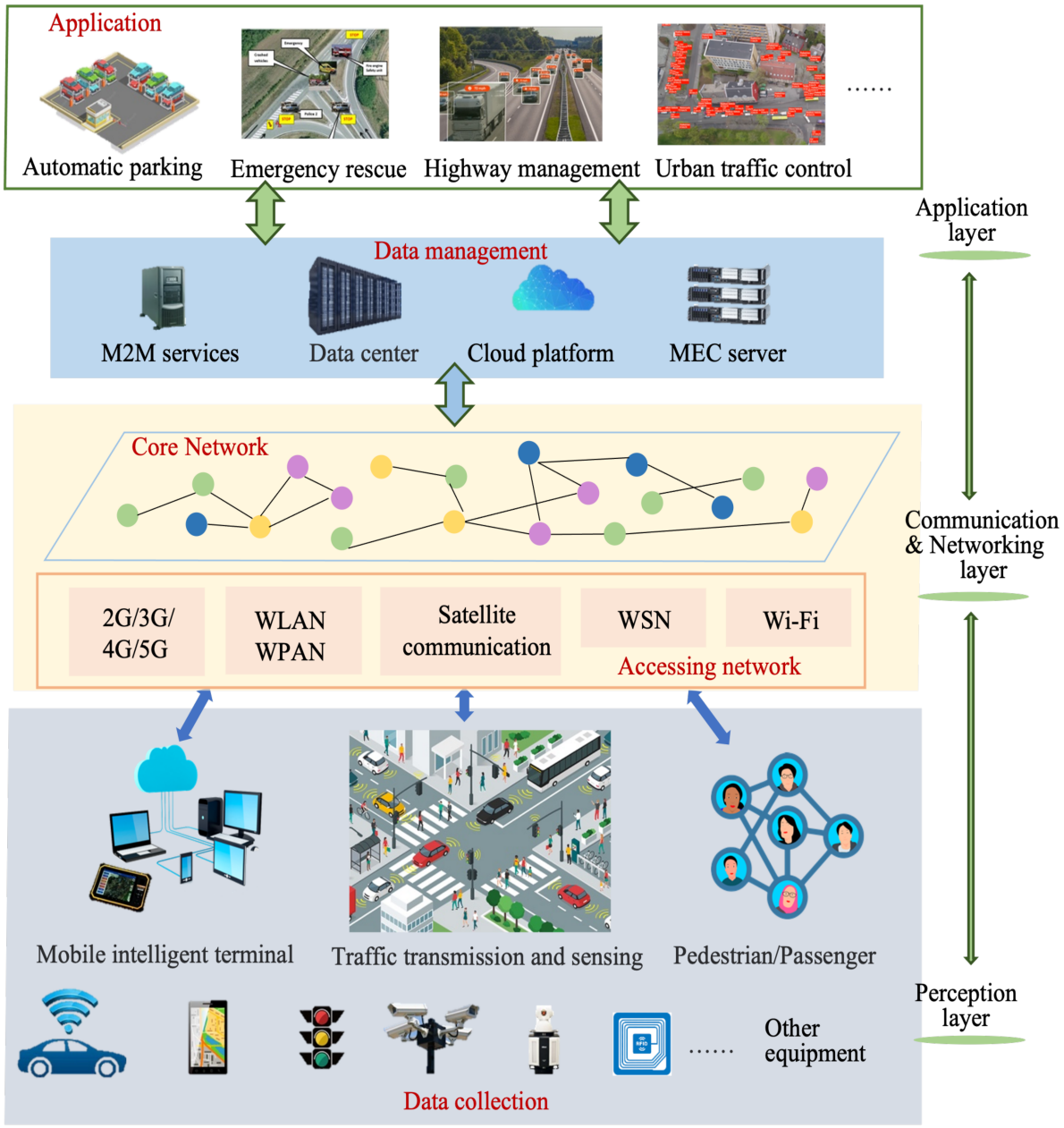

Figure 1.1: IoV Architecture.

1.2.4 IoV Communications

Various communication modes are used in vehicle networks, enabling road users to access and exchange data in real time. Among these, V2X communication plays a central role in allowing vehicles to interact with their environment, crucial for safety, traffic efficiency, and autonomous driving [8]. Key types of V2X communication are outlined below in Figure 1.2 adapted from [12].

- **Vehicle to Vehicle (V2V)** : Operates in ad hoc mode, enabling vehicles to directly transmit, receive, and exchange critical road information with one another. This includes data on traffic conditions, road accidents, and other relevant updates.
- **Vehicle to Infrastructure (V2I)** : Facilitates the exchange of crucial information between network infrastructure and vehicles. It allows vehicles to access data about road conditions and necessary safety measures by connecting to Road Side Units (RSUs) and external networks such as the Internet.
- **Vehicle to Road Side Unit (V2R)** : In this mode, vehicles interact with fixed roadside infrastructure to provide users with communication and information services. When transmitting information between vehicles, nearby cars communicate using the V2V technique, while more distant vehicles rely on V2R communication. This mode enhances overall vehicular network performance.
- **Vehicle to Personal Devices (V2P)** : Enables direct interaction between vehicles and vulnerable road users, such as pedestrians and cyclists, through personal devices like smartphones, tablets, or wearables. The system can issue real-time alerts to prevent collisions.

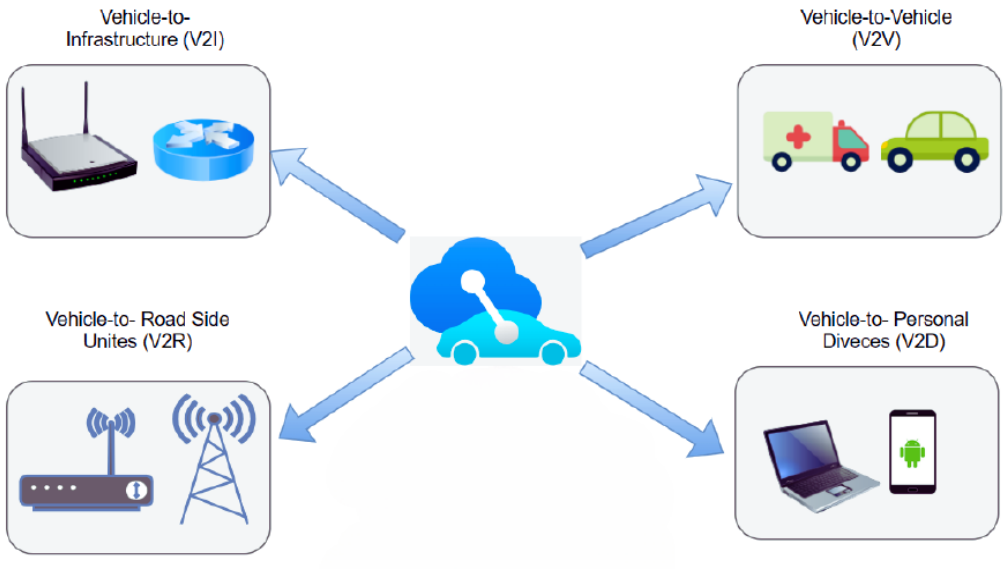

Figure 1.2: IoV Communications.

1.2.5 Security Challenges in IoV

IoV holds great promise for transforming transportation by integrating on board computers, Global Positioning System (GPS), and high speed communication technologies that enable real-time data sharing and automation [13]. However, this advancement also introduces significant security challenges. Unlike traditional networks, IoV's dynamic topology, infrastructure-less design, and time-sensitive data exchanges make it highly vulnerable to cyber threats.

Vehicles continuously exchange critical safety related information, so any compromise such as tampering or unauthorized access can have severe consequences [14]. Ensuring the confidentiality, authenticity, and integrity of these communications is therefore essential. To address these concerns, security mechanisms like encryption, digital signatures, secure broadcasting, and layered security architectures (e.g., Internet Protocol Security (IPsec)) are implemented to prevent eavesdropping, identity theft, and Denial of Service (DoS)attacks [15].

The emergence of autonomous and 5G-enabled vehicles further expands the attack surface, as increased connectivity and computational power provide more opportunities for adversaries to exploit vulnerabilities [16, 17]. To counter these threats, Misbehavior Detection System (MDS) have been proposed to detect suspicious data and prevent malicious activity [16].

Among the most dangerous threats facing IoV systems are APT, sophisticated, long-term cyberattacks where adversaries infiltrate networks stealthily, exfiltrate sensitive data, and continuously adapt their tactics to avoid detection. Unlike traditional attacks, APTs often combine multiple techniques and target systemic weaknesses, making them a critical concern for IoV security and resilience [18].

While APT behavior in IoV is often modeled through simulation (e.g., replay or Sybil attacks), these attack types exhibit many APT like characteristics such as stealth, persistence, and identity misuse central to our detection focus in Chapter 3.

Despite ongoing advancements, the rising complexity of cyber threats demands more adaptive and intelligent solutions for securing IoV systems[19].

1.2.6 Identity in IoV: Foundations and Challenges

1. **Definition and Role of Identity in IoV:** Identity in IoV refers to the unique representation of entities (vehicles, RSUs, users) within the network, enabling authentication, authorization, and accountability. It involves long-term identities, which are persistent identifiers (e.g., vehicle registration numbers, manufacturer-assigned IDs); short-term identities (pseudonyms), which are temporary identifiers used for privacy preservation; and cryptographic credentials, including public/private key pairs and certificates issued by Trusted Authority (TA)s or Certification Authority (CA)s [20].

   These identity types serve several critical functions:

   - **Authentication & Security:** Ensures that only legitimate entities (vehicles/RSUs) can participate in the network. Prevents spoofing, Sybil attacks (where one node pretends to be many), and unauthorized access.
   - **Privacy Protection:** Pseudonyms prevent long-term tracking of vehicles. Messages signed under different pseudonyms cannot be linked, preserving driver anonymity.

- **Accountability:** Authorities can resolve pseudonyms to long-term identities if needed (e.g., for legal investigations).
- **Cross-Region Mobility:** Vehicles crossing into foreign regions obtain new pseudonyms from local CAs to blend into the anonymity set of that region [21].

2. **Challenges in Identity Management in IoV**

   - **User Authenticity vs. Anonymity:** Ensuring a vehicle is genuine while hiding the driver's identity to preserve privacy is a fundamental contradiction. The system must distinguish between genuine and malicious nodes without revealing real identities.
   - **Real-Time Verification Constraints:** Verifying certificates and digital signatures introduces latency. In high-speed traffic, even minor delays in verification can have critical consequences.
   - **Storage Burden:**Vehicles must store multiple public/private key pairs or certificates. Storage becomes infeasible for high-mobility, low-resource nodes [22, 23].

3. **Challenges in Dynamic IoV Environment:**

   - **Node Mobility and Dynamic Topology:**Frequent network entry and exit make persistent authentication difficult due to shifting neighbors and topology.
   - **Interoperability Across Manufacturers:**Vehicles and RSUs may use different wireless technologies (Wi-Fi, Dedicated Short-Range Communications (DSRC), Long-Term Evolution (LTE), Near Field Communication (NFC)), making the integration of diverse communication standards into a single system complex.
   - **Connectivity Limitations:** Wireless channels in dense environments often become congested, and DoS attacks or channel interference can easily disrupt time-sensitive services.
   - **Scalability:**The growing number of vehicles leads to increased message traffic and a higher authentication load, requiring systems to scale without compromising latency or security.
   - **Real-Time Requirements:**Safety messages must be authenticated, processed, and acted upon within milliseconds, but complex cryptographic operations can delay their usability, making them unusable upon arrival [22, 23].

4. **Why Identity Matters in Intrusion and APT Detection**
   Identity-based attacks often exploit improper identity authentication or escalate identity privileges within a network. Traditional IDSs fail to detect such stealthy intrusions due to reliance on static signatures, high false positive rates in anomaly detection, and a general lack of awareness of who (identity) is acting abnormally[24].

   Using identity enables the IDS to model APT-like activities such as identity misuse over time, including identity switches and impersonation, as well as unexpected privilege escalation through trust manipulation[25].

1.3 Advanced Persistent Threats (APT)

In recent years, a new class of threats known as Advanced Persistent Threats has emerged. Originally used to describe cyberattacks against military organizations, the term has since expanded beyond the military domain, targeting a wide range of industries and government

entities [26]. In this section, we explore the nature of APT attacks, particularly in the context of IoV.

1.3.1 APT Definition

Although APTs have received significant attention in recent years, the term can sometimes appear vague, as experts may interpret it differently. In this work, we adopt the definition given in [27].

An APT, as the name suggests, differs from traditional cyberattacks in several ways. It is typically carried out by highly trained attackers who are often funded by nation states or large organizations to obtain critical information from targeted systems. Originally used in the military, the term "APT" is now widely used in cybersecurity to describe stealthy, resource-intensive, and goal-oriented attacks. APTs are defined by the combination of three key elements:

Advanced: The APT attackers can create sophisticated tools by combining several attack techniques and carrying out multi phase attacks.
Persistent: The APT attackers plan evasion strategies to avoid detection and are extremely persistent in their pursuit of the target. Typically, the APT uses a persistent method, such as the "low and slow" approach, to accomplish this.
Threat: To accomplish their objectives, the APT attackers target certain organizations with precision. They typically have the ability to damage an information system by destroying, disclosing, altering, or conducting denial-of-service attacks.

Initially associated with military and governmental systems, APTs now increasingly target emerging domains like vehicular networks, where persistent access can lead to traffic manipulation, surveillance, or large-scale disruption [26, 27, 28]

1.3.2 APT Characteristics in IoV

- **Learning from the environment:**APTs possess the ability to learn from the environment and to adapt the attack strategies accordingly to reduce detection chances.
- **Reconnaissance and Intelligence Gathering:**APTs aim to study the deployed defense system over a period of time until achieving sufficient reconnaissance to launch intelligent, high-impact cyberattacks at a large scale.
- **Targeting and Knowledge Acquisition:**APTs target specific system components and acquire useful knowledge about valuable assets and defense actions to carry out stealthy adaptive attacks that bypass detection.
- **Multi-Entity Targeting:**APTs may target multiple infrastructural elements (e.g., roadside units, traffic controllers, or specific vehicular components) at different magnitudes. This approach helps them prolong their activity while reducing the likelihood of detection.
- **Zero-Day Exploitation:**APTs will often exploit zero-day vulnerabilities within the system or network to increase their stealth.
- **Resource Intensity:** APTs require reasonable amounts of adversarial resources and sufficient reconnaissance of system operations to be materialized into lethal and long-lasting cyberattacks.

- **Identity Exploitation:**APTs often exploit identity-related behaviors such as credential theft, impersonation, and lateral movement through trusted relationships to escalate privileges and maintain long-term, unauthorized access within targeted systems [25, 28].

1.3.3 Stages of APTs in the IoV

The lifecycle of APT attacks includes multiple stages that demonstrate how persistent and stealthy these attacks are [29]. However, because IoV ecosystems are dynamic and complex, the APT lifecycle takes on unique characteristics in the context of IoV. The stages are shown in Figure 1.3 [28].

1. **Reconnaissance:** In the initial phase of an APT attack, attackers observe the network and gather intelligence about valuable targets by studying identity management systems, such as certificates and pseudonym rotation, and by analyzing the time schedules of traffic signal control algorithms used broadside units or signal controllers.

2. **Vehicle Compromise:** During this stage, attackers may physically access the victim's car or remotely compromise it through malware and privilege escalation. Once the vehicle is under control, it can function as a bot within the attacker's botnet, enabling harmful operations on the IoV network, including identity theft or misuse through certificate compromise and impersonation.

3. **Control data sensing and transmission:** In this stage, attackers compromise control data sensing and transmission in the IoV network by intercepting or modifying sensor readings and messages. This manipulation can inject false information or disrupt communication between vehicles and infrastructure, resulting in inaccurate data, degraded system performance, and potential escalation to more severe attacks.

4. **Launch Attack:** Compromised vehicles execute malicious acts by damaging sensors and message transmissions within the IoV network during the launch attack stage. IoV service performance is lowered as a result of this corruption, which interferes with the system's regular operations.

5. **Physical Impact:** The IoV network's components are physically impacted when data is corrupted during transmission. For instance, risky driving conditions, traffic jams, or even accidents can result from altered sensor data or interrupted communication. This stage highlights the real-world consequences of APT attacks in IoV, where cyberattacks directly affect physical systems and services.

6. **Observing and learning defense strategies:** Attackers continuously observe the environment, assess the impact of their actions, and learn the defense strategies deployed by the IoV system. To remain hidden and bypass Intrusion Detection Systems (IDS), they adapt their attack methods over time, including tactics such as pseudonym rotation to evade detection.

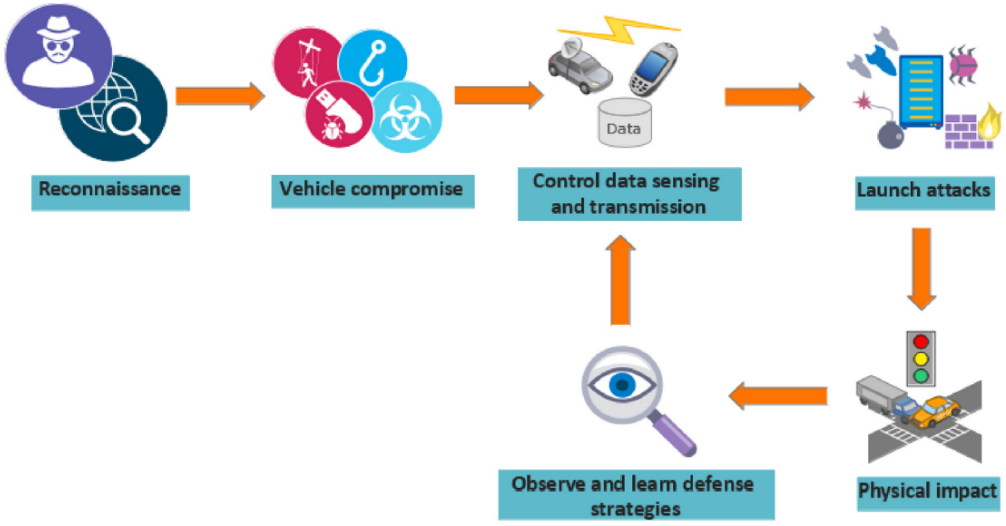

Figure 1.3: Stages of APTs in the IoV.

1.3.4 Identity-Based APT Attacks in IoV

Identity-based APTs in IoV are long-term, stealthy cyberattacks where attackers exploit identity systems like credentials and certificates to infiltrate vehicular networks, stay hidden, and move through the system undetected [30].

Attackers are motivated to exploit identities in IoV for several reasons. By assuming legitimate identities, they can maintain stealth and persistence, remaining undetected within the network for extended periods. Compromised identities also enable privilege escalation, allowing attackers to access higher-level functions or sensitive data that would otherwise be restricted. Additionally, manipulating identities can disrupt critical services by spreading misinformation, which may lead to traffic disruptions, safety risks, or accidents [31].

1.3.5 Common Attacks in IoV as APT Vectors

Although few cases have been found in vehicular communication, we cannot deny the possibility that APT-like attacks occur in cars as the automobile industry develops. Data fabrication or alteration threatening the integrity of the transportation system may be achieved through several forms of attacks, usually carried out during data transmission. When these techniques are executed persistently and adaptively, they take on APT characteristics. Examples include:

- **Sybil Attack:** A single vehicle uses multiple falsified identities to manipulate consensus or overload traffic control logic.
- **Replay Attack:** An attacker vehicle—such as a malicious node or a zombie controlled remotely—can repeatedly replay the same message to distort traffic conditions or trigger false congestion alerts. This becomes especially dangerous when tied to valid vehicle identities, allowing it to bypass basic verification.
- **Wormhole Attack:** Malicious vehicles transfer messages between distant locations to falsify topology and mislead infrastructure.
- **Masquerading:** The attacker impersonates another vehicle using stolen or spoofed credentials to access or disrupt systems.

- **GPS Spoofing:** Falsified coordinates mislead vehicle systems or RSUs, creating misrouting or route manipulation.
- **DoS/Distributed Denial of Service (DDoS)** Resource exhaustion used strategically to hide identity abuse or impair IDS visibility [20, 23, 28].

These are not standalone APTs, but when orchestrated to persist, evade, and adapt, they form a behavioral signature of APT-like activity.

1.3.6 APT effects in IoV

APT in IoV can have several significant effects, which are mentioned below:

- **Data Corruption and Manipulation:** Data corruption attacks can be launched via malicious or zombie vehicles by exploiting vulnerabilities in IoV networks and system components. For example, adversaries might deliberately alter their arrival times at intersections or manipulate their location data, which could compromise traffic signal controllers and lead to disastrous traffic congestion, effectively crippling the transportation network.
- **Resource Exhaustion:**Advanced Persistent Threats (APTs) can affect the availability of IoV services by launching resource exhaustion attacks that overwhelm critical components like Road-Side Units (RSUs). These attacks lead to Denial of Service (DoS), making RSUs unresponsive and disrupting real-time traffic coordination, communication, and intelligent transportation functions, ultimately compromising road safety and decision-making.
- **Compromised Decision-Making:**The performance and reliability of IoV applications managed by the Traffic Management Center (TMC) can be seriously affected when corrupted or manipulated data leads to suboptimal decisions such as incorrect routing, poor signal timing, or failure to detect congestion, ultimately degrading traffic optimization and transportation system intelligence [28].
- **Trust Erosion and System Confusion:** Attacks such as impersonation and Sybil can lead to trust erosion within IoV systems, as malicious vehicles use fake identities to spread false information. This happens because repeated exposure to falsified or conflicting data undermines the credibility of the network, making it difficult for vehicles and infrastructure to distinguish between legitimate and malicious entities. As a result, this manipulation causes system confusion, leading to poor traffic management decisions and potential safety risks [32].

1.3.7 Real-World APT-Like Scenario: Jeep Cherokee Hack

A relevant real-world incident that exemplifies the behavior of an APT in an IoV environment is the 2015 Jeep Cherokee cyberattack conducted by cybersecurity researchers Charlie Miller and Chris Valasek.

The vehicle's cellular-enabled infotainment system, Uconnect, exposed a vulnerable surface over the public internet via the Sprint network. By scanning IP ranges, the attackers identified a vehicle online and remotely accessed its Head Unit (HU), bypassing authentication mechanisms and exploiting a zero-day vulnerability. They injected malicious firmware, establishing a persistent backdoor that remained active even after system restarts.

From this foothold, they escalated access and pivoted laterally into the Controller Area Network (CAN) bus, the vehicle's internal communication system. Using crafted CAN messages, they issued direct commands to critical Electronic Control Unit (ECU)s, including those responsible for braking, acceleration, and steering. This enabled full remote control of the vehicle while it was in motion, without driver awareness.

Though conducted as a controlled security demonstration, the structure of the attack mirrors the phases of a real APT:

- **Initial compromise**: via cellular interface.
- **Persistence**: through firmware-level access.
- **Lateral movement**: across vehicular subsystems.
- **Stealthy control**: of mission-critical vehicle functions.

This event forced Fiat Chrysler to recall approximately 1.4 million vehicles, underscoring the severity of cyber threats targeting IoV systems. It has since been cited as a foundational case in automotive cybersecurity, demonstrating that advanced, identity-abusing, persistent threats are not only feasible but potentially catastrophic in vehicular networks[33].

This scenario validates the motivation behind developing intelligent and context-aware intrusion detection models, especially those capable of identifying identity-based APTs within vehicular communication flows, as proposed in this thesis.

1.3.8 Defenses Against APTs in IoV

1. **Intrusion Detection Systems:** Intrusion Detection and Prevention Systems (IDPS) in IoV focus on monitoring identity-related behaviors such as credential use and pseudonym changes—to detect threats [34]. At the same time, IDSs continuously analyze network traffic and system behavior to identify sophisticated, long-term attacks like APTs, using advanced adaptive methods beyond traditional detection techniques.

2. **AI techniques** Artificial Intelligence (AI)-powered machine learning techniques enhance Intrusion Detection Systems in vehicular networks by detecting identity-related anomalies such as credential misuse, pseudonym changes, and other unusual behaviors [35]. Using advanced models like Long Short-Term Memory (LSTM) and RNN, they learn normal traffic patterns to identify stealthy threats like APTs in real-time, continuously adapting to the dynamic Internet of Vehicles environment.

3. **Game Theory** Game theory has emerged as a powerful modeling framework for analyzing the dynamic interactions between attackers and defenders in (IoV) environments, especially in the context of APTs. In particular, a Bayesian Stackelberg game is applied, where the IoV defense system (e.g., the Intrusion Detection System at the Traffic Management Center) acts as the leader, and APT attackers as followers. As proposed in [28], the defender randomizes which Roadside Units (RSUs) to monitor, making it harder for APTs to exploit predictable defense patterns. This approach allows the defender to anticipate and counteract stealthy, long-term attacks by optimizing monitoring strategies under uncertainty and reducing the chances of detection evasion.

1.4 Intrusion Detection System (IDS):Challenges in IoV

Intrusion Detection Systems (IDS) aim to identify malicious behaviors in network or system activity [36]. While effective in traditional networks, IDS face serious limitations in Internet of Vehicles (IoV) due to dynamic topologies, real-time constraints, and identity-based threats such as Sybil and replay attacks[37].

Traditional IDS, including signature-based and anomaly-based models, often rely on static rules or short-term pattern monitoring [38, 39]. This makes them poorly suited for APT attacks that unfold slowly and involve credential misuse or pseudonym abuse. Additionally, distributed IDS architectures, while more compatible with IoV, still struggle with scalability, high false positives, and limited insight into identity behaviors[40].

These challenges underline the need for AI-based intrusion detection, specifically deep learning techniques capable of modeling complex, temporal, and identity- related attack patterns in IoV [37]. This is the direction explored in Chapter 3.

1.5 Artificial Intelligence (AI) in Cybersecurity

1.5.1 Overview of AI in Cybersecurity

AI refers to the development of computer systems capable of performing tasks that typically require human intelligence, such as reasoning, learning, and decision-making. The term was introduced by John McCarthy in 1956 to define the science of building intelligent machines and programs. AI is categorized into Narrow AI, which handles specific tasks like voice recognition or spam detection, and General AI, a theoretical model capable of performing any intellectual task a human can [41, 42].In cybersecurity, particularly in vehicular networks, AI is pivotal in automating detection, analyzing patterns to discover zero-day exploits, filtering false alarms, and reacting adaptively to evolving threats. Its applications range from intrusion detection and malware classification to automated mitigation and threat prediction. AI-driven models can process large-scale traffic and behavioral data to uncover identity-based APTs, which are often missed by static or rule-based systems [43, 44, 45].

1.5.2 AI Techniques for Intrusion Detection Systems

AI has revolutionized IDSs by enabling automated, adaptive, and intelligent threat detection. Unlike traditional signature-based IDS, AI-powered systems can learn from data, detect novel attacks, and reduce false alarms. Below, we explore the key AI techniques used in modern IDS, with clear definitions, use cases, and references to research.

1. **Machine Learning (ML):** Machine Learning allows IDS to automatically detect known malware signatures, phishing attempts, or brute-force attacks, by learning from historical network traffic data. ML techniques can be broadly categorized into the following:

    a. **Supervised Learning:** Trains models using labeled datasets (normal vs. malicious traffic) to classify future threats; the algorithm learns patterns from past attacks and applies them to new data. For example, we have:

    - **Decision Trees / Random Forest:** Builds a flowchart-like model to classify traffic, it's highly interpretable and effective for detecting known attack patterns.

- **XGBoost:** a powerful gradient boosting algorithm known for its accuracy and efficiency in classifying network traffic and detecting attacks. and others like Support Vector Machine (SVM) and Naïve Bayes...)[46, 47, 48, 49]

b. **Unsupervised Learning:** Detects unknown attacks without labeled data by finding anomalies in network behavior (Identifying zero-day exploits or insider threats where no prior attack data exists) using clustering to group similar traffic, flagging outliers as potential threats. Key Algorithms:

- K-Means Clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and Hierarchical Clustering.

c. **Reinforcment Learning (RL):** AI agents learn optimal defense strategies through trial and error, receiving rewards for correct detections. The system interacts with live traffic, adjusting detection rules to maximize accuracy. Some of the main algorithms used are Q-Learning, Deep Q-Networks (DQN)[48]. Figure 1.4 is adapted from [50].

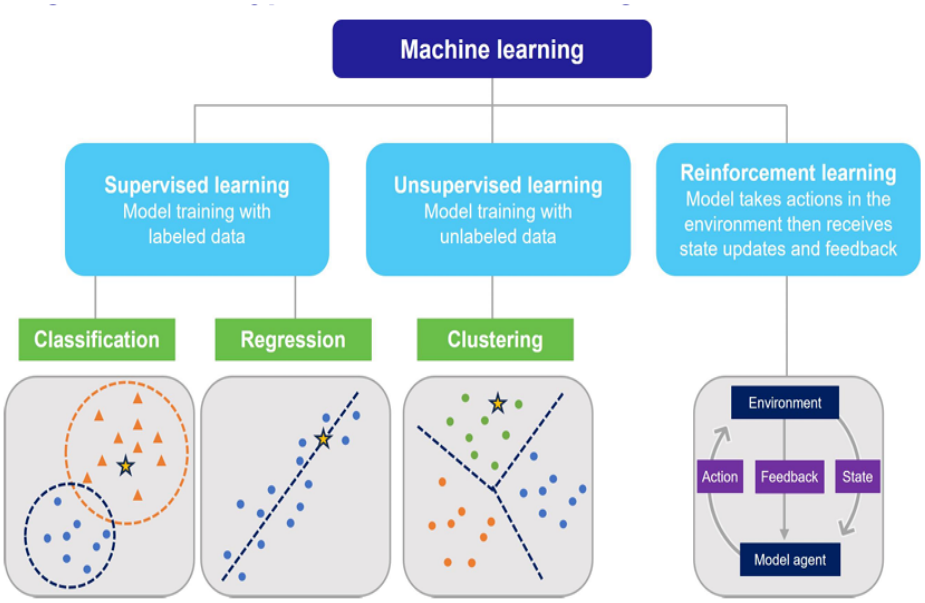

Figure 1.4: ML paradigms and typical algorithms used in Intrusion Detection Systems

2. **Deep Learning (DL):** Deep Learning excels at analyzing complex, high-dimensional data such as raw packet captures. Its techniques are divided into :

a. **Neural Networks (NN):** Inspired by the human brain, neural networks process data through interconnected layers to detect subtle attack patterns. These layers include an input layer that receives raw network traffic, hidden layers that extract meaningful features, and an output layer that classifies the traffic as normal or malicious. Common types of NN include:

- **Feedforward Neural Network (FNN):** The simplest architecture, where data flows in one direction from input to output, suitable for basic attack classification. .
- **Backpropagation Networks:** Enhance detection accuracy by adjusting neuron weights based on errors, allowing the model to learn from past attack patterns [46, 48, 49].

b. **Convolutional Neural Network (CNN)s:** Specialized for analyzing spatial data such as network traffic patterns. They consist of convolutional layers that automatically extract features like unusual packet sizes or timing anomalies, followed by pooling layers that reduce data complexity for efficient processing. CNNs are effective at detecting malware hidden in encrypted traffic by recognizing anomalous packet structures, and at identifying DDoS attacks by analyzing traffic flow patterns over time [47, 48].

c. **Recurrent Neural Networks (RNNs):** Designed for sequential data, making them effective for detecting stealthy, long-term attacks that unfold over time, such as Advanced Persistent Threats (APTs). By retaining information from previous inputs, RNNs can track evolving patterns in network traffic, identifying multi-stage intrusions like data exfiltration or persistent breaches[47, 48].

d. **Autoencoders:** A form of unsupervised deep learning, can also support intrusion detection by learning compressed representations of normal traffic and flagging deviations as potential anomalies. They consist of an encoder that compresses input data and a decoder that reconstructs it, minimizing reconstruction error. Variants like Sparse, Denoising, and Variational Autoencoders enhance robustness and generalization. Though not used in this study's implementation, autoencoders remain relevant in detecting subtle, previously unseen attack behaviors in IoV networks[51]

3. **Hybrid AI Models:** To improve accuracy and adaptability, modern IDS often combine multiple AI techniques, leveraging the strengths of each approach for a more comprehensive defense. For example, machine learning and deep learning can work together, where a Random Forest model quickly classifies traffic, and a CNN further analyzes complex patterns to refine detection. Similarly, supervised and unsupervised learning can be combined, where supervised models identify known threats, while clustering methods detect anomalies that may indicate new or evolving attacks.

   By integrating these approaches, hybrid models create robust, multi-layered security frameworks that enhance threat detection while maintaining flexibility in dynamic environments[47, 48].

4. **Explainable Artificial Intelligence (XAI):** Many AI-based IDS function as "black boxes," making it diffi- cult to understand how they reach their decisions. This lack of transparency can hinder trust and make it challenging for security teams to act confidently on AI-generated alerts. Explainable AI (XAI) addresses this issue by providing insights into the reasoning behind threat detection. Two widely used XAI methods are:

   - **SHAP (Shapley Additive Explanations)** that helps determine how much each feature (e.g IP address, port number) contributes to a detection, offering a clearer view of why an alert was triggered.
   - **LIME (Local Interpretable Model-Agnostic Explanations)** that simplifies complex models by generating interpretable approximations, making AI-driven decisions eas- ier to understand.

   By enhancing transparency, XAI not only builds trust in AI-powered IDS but also helps analysts fine-tune detection rules, reduce false positives, and improve overall threat detection. response[49].

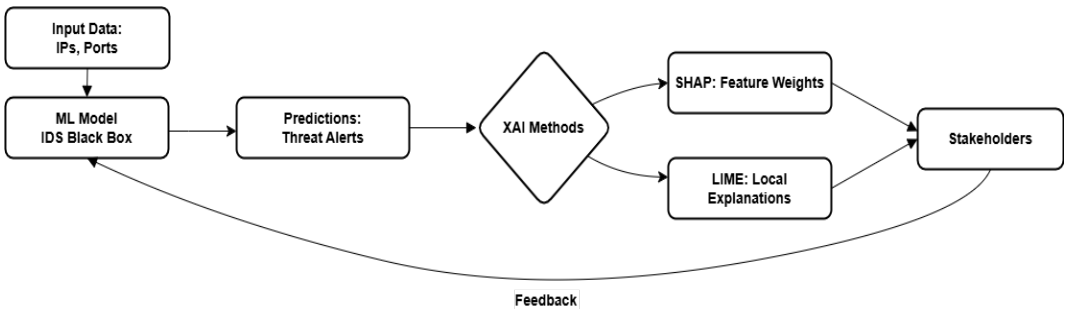

Figure 1.5: How XAI Explains AI Threat Detection: From Black Box to Actionable Alerts.

1.5.3 Deep Learning Use in Intrusion Detection

CNNs have been widely adopted in IDS due to their ability to extract spatial patterns from network traffic data. These models process raw network packets, identifying differences between normal and malicious traffic. Research has shown that CNN-based IDS outperform traditional ML methods in classifying various cyber threats, including denial-of-service (DoS) attacks, brute-force login attempts, and malware intrusions. However, CNNs struggle with sequential dependencies, making them less effective against attacks that unfold over time [52, 53].

To address this, Recurrent Neural Networks (RNNs) and their advanced variant, Long Short-Term Memory (LSTM) networks, have been widely adopted. While RNNs process sequential data by retaining short-term memory, LSTMs enhance this with long-term dependency tracking. This makes them ideal for detecting identity-based threats that unfold gradually. By analyzing network behavior across sequences, these models identify subtle anomalies missed by conventional methods.

Other RNN variants, such as Gated Recurrent Units (GRUs), also offer efficient alternatives with comparable performance and reduced computational complexity [54, 53].

1.6 Conclusion

This chapter has established the foundational context for understanding security challenges in the Internet of Vehicles (IoV), with a particular focus on identity-based threats and Advanced Persistent Threat (APT)-like behaviors. We explored how IoV's architecture, communication models, and reliance on trust systems expose it to subtle forms of identity misuse such as Sybil, impersonation, and replay attacks.

The limitations of traditional Intrusion Detection Systems (IDS)—including static signature reliance and a lack of identity-awareness—make them ill-suited for detecting these evolving threats. To overcome these shortcomings, deep learning (DL) models, particularly those capable of temporal sequence modeling, have emerged as powerful alternatives.

Importantly, this work shifts the focus from general anomaly detection to identity-centric behavioral analysis. By interpreting replayed or falsified identities as signs of long-term deception, even without access to true identity labels, we lay the foundation for an adaptive detection framework.

The next chapter builds on this foundation by surveying deep learning models used in IDS, setting the stage for the hybrid model proposed in Chapter 3.

Chapter 2

Literature Review and Deep Learning Approaches in IoV Security

2.1 Introduction

This chapter reviews existing approaches to intrusion detection in vehicular networks, with a special focus on sequential deep learning models. Due to the time-series nature of vehicular communication, RNNs and their variants have gained attention for their ability to model temporal dependencies in network traffic. Traditional machine learning methods often fall short in this area, which motivates the growing use of deep learning—especially architectures like LSTM, Gated Recurrent Unit (GRU), and hybrid models. The chapter begins with an overview of RNNs and their main variants, followed by a structured review of related works grouped into three categories: pure RNN models, hybrid CNN-RNN architectures, and multi-RNN or autoencoder-based models. A comparative analysis then summarizes and contrasts the key findings across these approaches.

2.2 Convolutional Neural Networks

2.2.1 Definition and Core Architecture of CNNs

CNNs are a class of deep neural networks specifically designed to automatically and adaptively learn spatial hierarchies of features from input data through the use of convolutional operations. Originally inspired by the structure and function of the animal visual cortex, CNNs have demonstrated remarkable success in domains that involve grid-like data structures, such as image processing and signal analysis[55].

A typical CNN consists of five main layers arranged in a sequential pipeline. It begins with the input layer, which receives the raw data, such as numerical vectors or matrices. This is followed by one or more convolutional layers, where learnable filters slide over the input to extract local spatial features. Next, pooling layers are applied—most commonly max pooling to downsample the feature maps, reducing their dimensionality and helping the model become more robust to spatial variations. The fully connected layer then flattens the feature maps and connects all neurons to combine the extracted information for final decision making. Lastly, the output layer, often implemented with a Softmax activation function, generates the final prediction as a probability distribution over the possible classes. This structure enables CNNs to effectively learn spatial hierarchies from input data, even when the original input is non-image-based, such as sensor or network traffic data[56]. Figure 2.1 adapted from [57] illustrates the basic architecture of a CNN, highlighting the sequential flow through the input, convolutional, pooling, fully connected, and output layers.

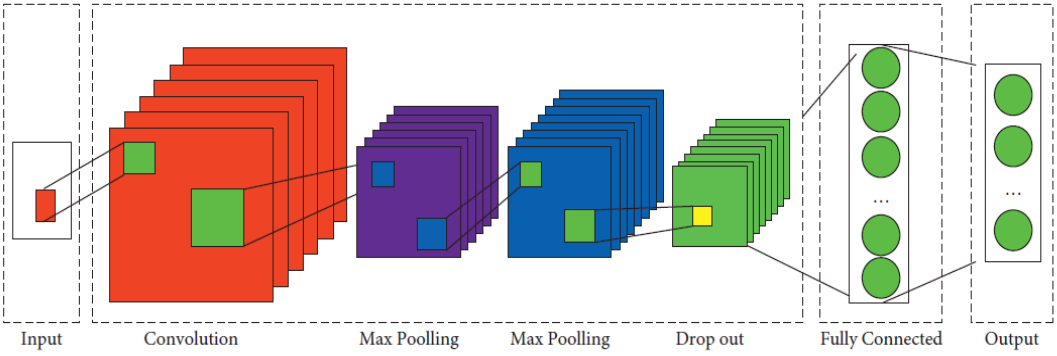

Figure 2.1: CNN architecture.

Although CNNs are traditionally associated with image based data, they have been successfully adapted to domains such as cybersecurity and intrusion detection in IoV. In these contexts, structured input data such as network traffic records, vehicle telemetry, or CAN bus message fields (e.g., IDs, timing, and payloads)—can be reshaped into one-dimensional or two-dimensional arrays that preserve spatial relationships between features. One-dimensional CNNs process flat feature vectors, while two dimensional CNNs interpret the same data as matrices, often improving spatial pattern learning and computational efficiency. A notable extension of the 1D CNN architecture is the TCN, which is specifically designed for sequential data and uses dilated causal convolutions to model long-range temporal dependencies.

This flexibility allows CNNs and their variants to learn local patterns and dependencies that may indicate identity-based or behavioral anomalies. These extracted features can either be classified directly or passed to temporal models such as LSTM or GRU for further sequence modeling, enabling the detection of subtle and persistent threats in vehicular networks[55].

2.3 Recurrent Neural Networks

2.3.1 Definition and Architectures

RNNs are a class of deep learning models specifically designed to process and analyze sequential data. Unlike traditional feedforward neural networks, RNNs have an internal memory that allows them to retain information from previous inputs, enabling them to model temporal dependencies within data sequences. This characteristic makes RNNs particularly effective for tasks where the order and context of data points are essential, such as natural language processing, speech recognition, and time series forecasting[58, 59].

The fundamental architecture of RNNs consists of an input layer, a hidden layer, and an output layer as illustrated in Figure 2.2. The key distinction from feedforward neural networks lies in the presence of recurrent connections, which allow information to circulate within the network across time steps.

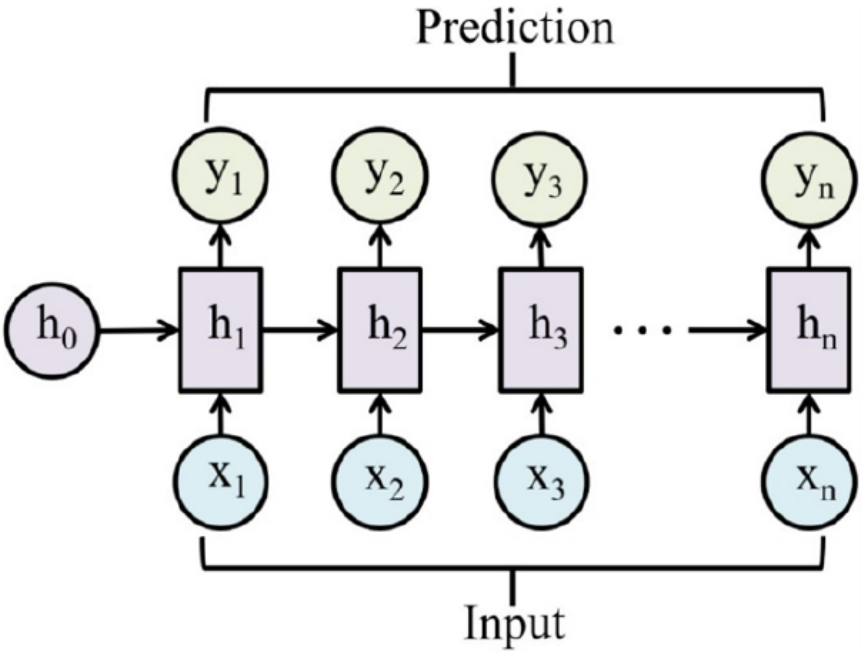

Figure 2.2: Fundamental RNN Architecture.

At each time step t, the RNN receives an input vector x_t and updates its hidden state h_t according to the following equation:

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.1}$$

Where W_{xh} denotes the weight matrix connecting the input layer to the hidden layer, while W_{hh} represents the weight matrix associated with the recurrent (hidden-to-hidden) connections. The term b_h is the bias vector, and σ_h refers to the activation function, which is typically the hyperbolic tangent function (tanh) or the Rectified Linear Unit (ReLU). The output at each time step t is computed as follows:

$$y_t = \sigma_y(W_{hy}h_t + b_y) \tag{2.2}$$

Where W_{hy} represents the weight matrix connecting the hidden layer to the output layer, b_y is the corresponding bias vector, and σ_y denotes the activation function applied at the output layer[58]. This architecture enables RNNs to effectively learn from temporal patterns by leveraging both current and past information across sequences.

2.3.2 Role in analyzing sequential data

RNNs are a class of neural architectures specifically designed for processing sequential data. Unlike feedforward networks, RNNs maintain a dynamic hidden state that evolves over time, enabling the model to incorporate both current inputs and contextual information from previous time steps. This makes RNNs particularly effective for modeling temporal dependencies in time-series tasks. They are capable of learning:

- **Short-term dependencies:**Recent elements such as predicting the next character in a word.
- **Long-term dependencies:**More distant elements such as a subject-verb agreement in a sentence.

However, standard RNNs are often limited in capturing long-range patterns due to the vanishing gradient problem during training, which causes gradients to diminish exponentially as they are propagated backward through time, making it difficult for the network to learn dependencies from earlier time steps. To address this, advanced variants such as LSTM networks and GRU were introduced, which are specifically designed to better capture long-term dependencies in sequential data.[60, 61].

RNN-based models have seen broad application across domains:

- In network anomaly detection, LSTM/GRU models identify temporal irregularities in telemetry streams [62].
- In autonomous driving, they predict lane-change intentions based on sequences of sensor data [63].
- In activity recognition, Bidirectional LSTMs learn movement patterns from wearable devices [64].

2.3.3 Relevance of RNNs for Temporal Anomaly Detection in IoV and Identity-Based Attacks

Vehicular networks generate vast amounts of spatiotemporal traffic data, characterized by both sequentiality and periodicity. Sequentiality refers to the chronological order of data collection, where each data point is timestamped later than the previous one, while periodicity encompasses recurring patterns such as daily traffic cycles. Capturing these temporal characteristics is vital for modeling normal behavior and detecting anomalies.

Traditional machine learning algorithms often fall short in this domain due to their reliance on manual feature engineering and their limited capacity to model temporal dependencies. For example, RNNs especially LSTM and GRU variants excel at learning from sequences, making them highly suitable for vehicular time-series modeling tasks such as traffic prediction and anomaly detection[65].

This capability becomes especially valuable in detecting identity-based APTs, such as:

- Replay attacks, where previously recorded messages are reintroduced, creating inconsistencies like outdated or duplicate timestamps[66].
- Sybil attacks, which inject coordinated waves of fake identities that disrupt natural vehicle behavior with unnatural timing patterns[67].

These attacks create small but noticeable timing irregularities that traditional rule-based systems often miss. RNNs can learn normal timing patterns, making them well-suited to detect these anomalies [68, 69]. This makes them a strong choice for improving intrusion detection in Intelligent Transportation System (ITS)[70, 71].r

2.3.4 Introduction to RNN variants

RNN architectures differ widely in their structural design—some incorporate internal recurrence within individual units, while others rely on external recurrence across multiple layers. These architectural differences influence how effectively the network can learn and model sequential patterns, ultimately shaping their suitability for various types of tasks.

1. **LSTM (Long Short-Term Memory):** An advanced variant of Recurrent Neural Networks (RNNs), introduced by Hochreiter and Schmidhuber in 1997 [72] to solve the vanishing gradient problem found in traditional RNNs. LSTMs are designed to better capture long-term dependencies in sequential data by using a system of gates to regulate the flow of information.

   The architecture of a LSTM network—illustrated in Figure 2.3, adapted from [73]—is centered around the concept of memory cells that maintain an internal state over time through the use of gating mechanisms. Each LSTM cell contains three gates: the **input gate**, **forget gate**, and **output gate**, which regulate the cell state and hidden state.

   These gates determine how much of the current input to consider, how much of the past information to retain or discard, and how much of the internal memory to pass forward. This gate-based structure enables LSTMs to learn and preserve important information over long time intervals, addressing the limitations of standard RNNs.

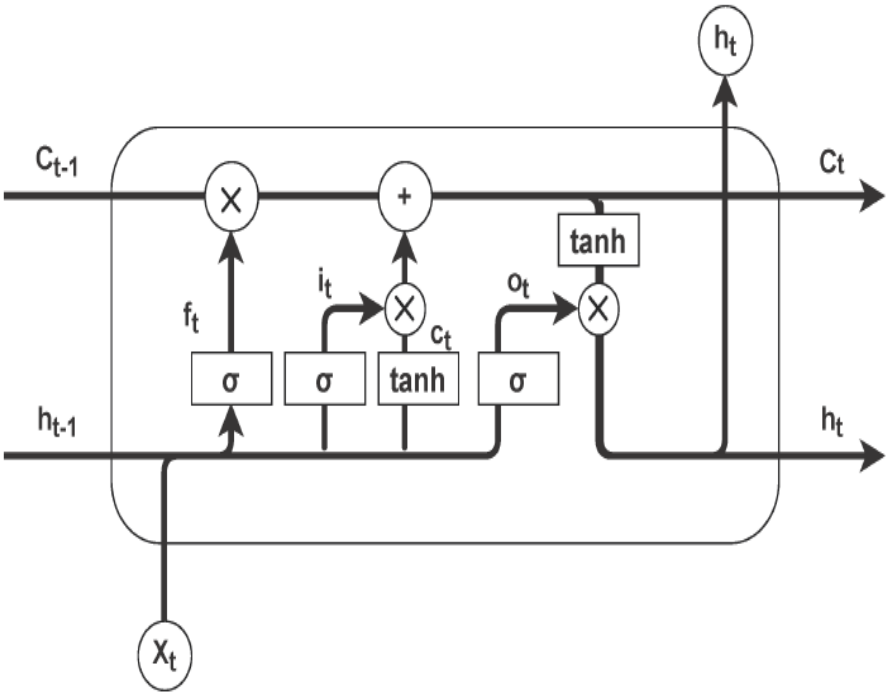

Figure 2.3: Architecture of the LSTM network.

2. **Stacked LSTM:** This architecture consists of multiple LSTM layers arranged hierarchically, where the output from one layer serves as the input to the next, as shown in Figure 2.4—adapted from [73]. This layered structure enables the model to learn increasingly abstract and hierarchical representations of temporal data, thereby improving its ability to capture complex patterns and long-range dependencies.

Figure 2.4: A stacked LSTM.

The vertical stacking of LSTM layers allows each layer to process and refine the temporal information extracted by the previous one. As data moves through the stacked layers, the model builds deeper understanding of the underlying temporal dynamics, which is particularly useful for tasks involving high-level sequence reasoning.

3. **Bidirectional Long Short-Term Memory (BiLSTM):** This model enhances the traditional LSTM by processing the input sequence in both forward and backward directions. The bidirectional structure allows the model to consider both past and future contexts at each time step, leading to a richer and more informative understanding of the data sequence.

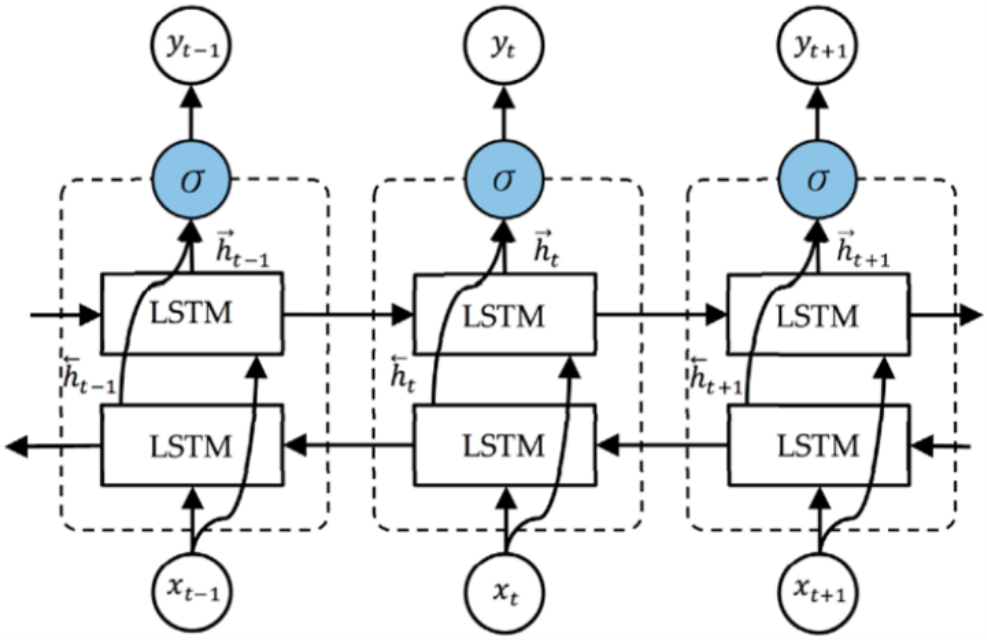

Figure 2.5: Architecture of the BiLSTM network.

As illustrated in Figure 2.5—adapted from [73]—the BiLSTM maintains two separate hidden states for each time step: one that processes the sequence from the beginning to the end, and another that processes it in reverse. The outputs from both directions are then combined to produce a more comprehensive representation of the input.

This architecture is particularly effective in tasks where context from both previous and upcoming elements is crucial, such as language modeling, anomaly detection, and behavioral sequence analysis. In the context of vehicular networks, BiLSTMs are especially useful for detecting identity-based APTs such as Sybil and Replay attacks, which introduce subtle temporal inconsistencies. By analyzing data in both directions, BiLSTMs are better equipped to uncover deviations from normal spatiotemporal communication patterns, enhancing the detection of such stealthy threats.

4. **GRU (Gated Recurrent Units):** GRUs are a simplified variant of LSTMs, proposed by [68], designed to address the vanishing gradient problem while reducing architectural complexity. Unlike LSTMs, GRUs merge the input and forget gates into a single *update gate* and combine the cell state and hidden state into one. This results in fewer parameters and makes GRUs computationally more efficient.

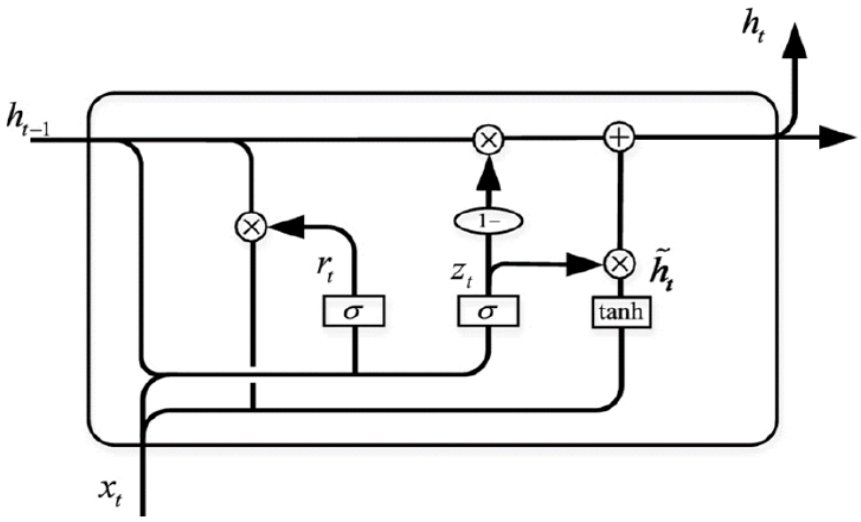

Figure 2.6: Architecture of the GRU network.

As illustrated in Figure 2.6—adapted from [73]—the GRU architecture includes two gates: the **update gate**, which determines how much of the past information to carry forward, and the **reset gate**, which decides how much of the previous state to forget. Like LSTMs, GRUs maintain internal recurrence and update the hidden state over time to model temporal dependencies in sequential data.

Despite their simplified structure, GRUs retain the ability to capture long-term dependencies and have demonstrated performance comparable to LSTMs across a wide range of sequence modeling tasks. Their efficiency and lower computational cost make them particularly well-suited for applications with limited processing resources [74].

5. **Stacked GRU:** To enhance the model's ability to learn complex temporal relationships, multiple GRU layers can be stacked to form a *stacked GRU* architecture. As shown in Figure 2.7, the output of one GRU layer serves as the input to the next, allowing the network to learn hierarchical representations of sequential data.

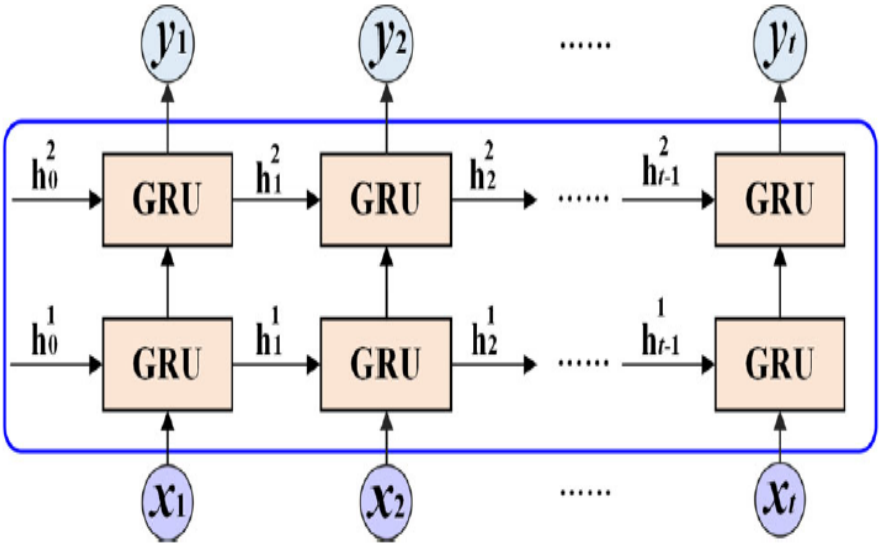

Figure 2.7: The structure of stacked GRU.

Lower layers in the stack typically focus on capturing short-term dependencies, while higher layers extract more abstract and long-range patterns. This deep structure increases the representational capacity of the model and is effective for tasks involving intricate time-based behaviors [75].

6. **BiGRU (Bidirectional GRU):** BiGRU extends the standard GRU architecture by processing sequences in both forward and backward directions. This bidirectional approach enables the model to use context from both the past and the future at each time step, resulting in a more complete understanding of the sequence.

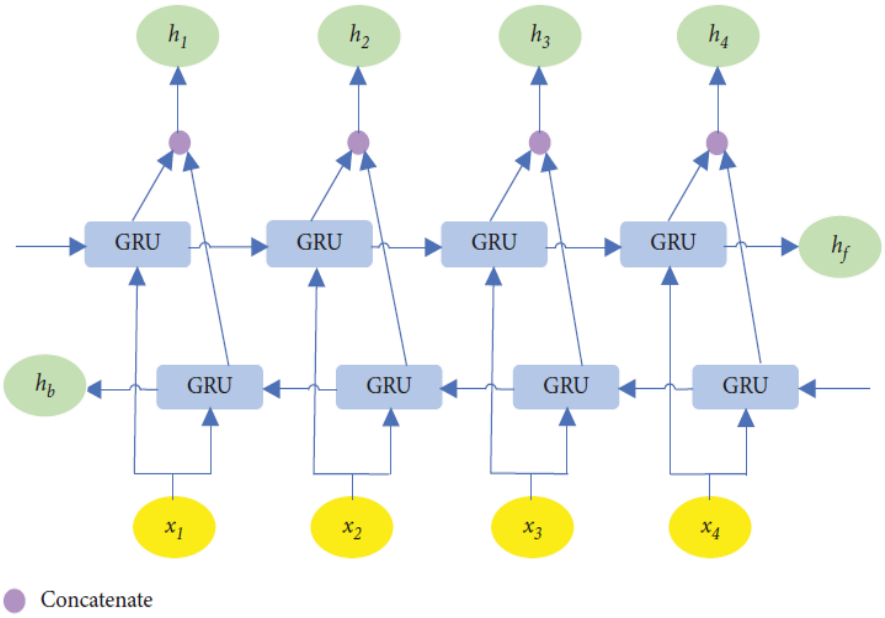

Figure 2.8: Architecture of BiGRU network.

As illustrated in Figure 2.8—adapted from [69]—each input passes through two parallel GRU layers: one processes the sequence from start to end, while the other processes it in reverse. The outputs from both directions are then combined to form the final hidden states. This structure enhances the model's ability to capture dependencies across the entire sequence and improves performance in tasks that require a full contextual understanding, such as text analysis, traffic anomaly detection, and behavior modeling [76].

While LSTM and GRU are among the most widely used variants of recurrent neural networks due to their effectiveness in handling long-term dependencies, several other architectures—such as peephole LSTM, echo state networks, and independently recurrent neural networks—have been proposed to address specific limitations or improve performance in certain contexts [58].

In summary, the various RNN architectures—from standard RNNs to LSTM, GRU, and their bidirectional or stacked versions—offer unique strengths in modeling temporal dependencies. Their ability to capture complex spatiotemporal patterns makes them particularly suitable for identifying anomalies in vehicular networks, especially those caused by identity-based APTs. In the following section, we explore how these models have been employed in the literature.

2.4 Related works

Several recent works have explored different RNN architectures either standalone or in combination with other deep learning models to enhance detection accuracy and improve security analysis in vehicular networks. This section reviews selected studies grouped into three main categories:

2.4.1 Pure RNN-Based Models

Studies in this category focus on using standard RNN variants such as LSTM and BiLSTM without additional neural components. For example, in [77], the authors used a standard LSTM to detect Sybil attacks in electric vehicle (EV) networks. Their model worked in a self-learning loop and tracked how traffic data changed over time to find both known and new attacks. They used simulated data from SUMO and NS2, with training data collected in a safe environment. The model detected Sybil attacks with about 95% accuracy and had a 5% false detection rate.

In another study, [78] used a BiLSTM model with an Attention mechanism to detect anomalies in CAN bus traffic inside vehicles. The BiLSTM captures patterns in both forward and backward directions, while the Attention helps the model focus on important features and speeds up training. They tested it on the HCRLab Public Dataset and the Korean Competition Dataset, both collected through the OBD-II port. The model performed well, detecting replay attacks with 92.8% accuracy, DoS with 97.9%, and fuzzing with 95.8%.

2.4.2 Hybrid CNN-RNN Architectures

Some studies combine convolutional layers with RNNs to capture both spatial and temporal features in vehicle data. For example, [79] proposed a hybrid model for IoV that uses a Temporal Convolutional Network (TCN) to extract spatial features, an LSTM to capture time-based patterns, and a self-attention mechanism to adaptively combine features. A Multilayer Perceptron (MLP) handles the final classification. Tested on the NSL-KDD and UNSW-NB15 datasets, the model reached 98.68% and 96.34% accuracy showing the effectiveness of combining CNN and RNN for spatio-temporal intrusion detection.

Another study [80] proposed a hybrid model that combines Graph Convolutional Networks (GCN), BiLSTM, and an attention mechanism to detect abnormal traffic in IoV. GCNs capture spatial relationships, BiLSTM learns time-based patterns in both directions, and attention helps highlight features from underrepresented classes. Tested on NSL-KDD and CICIDS2017, the model achieved over 94% F1-scores and detection rates of 95.87% and 94.12%, with low false positives. This shows that combining GCNs, RNNs, and attention works well for handling imbalanced IoV traffic data.

Authors in [81] proposed a hybrid intrusion detection model for VANETs that combines a Self-Attention BiLSTM (SA-BiLSTM) with a Cascaded CNN (CCNN), optimized using a Multi-variant Gradient-Based Optimization (MV-GBO) algorithm. The CCNN captures layered spatial features, while the SA-BiLSTM learns complex temporal patterns with attention. Tested on KDD-CUP99, ToN-IoT, and VeReMi, the model achieved high accuracy—99% on general datasets and 98.6% on VeReMi outperforming standard CNN-LSTM models.

Authors in [82] introduced CANintelliIDS, a hybrid intrusion detection model for CAN bus networks that combines CNN with an Attention-based GRU (AGRU). The CNN extracts spa-

tial features from traffic, while the AGRU captures time-based patterns and uses attention to focus on important inputs. Tested on a real-world dataset of 4.6 million CAN messages from a KIA Soul, the model achieved 94.06% accuracy and a 93.79% F1-score, outperforming models like CNN-LSTM, RNN, and SVM. It was especially effective against impersonation attacks, with a 94.01% F1-score, showing the value of using CNN-RNN hybrids with attention in detecting complex intrusions.

The paper [83] proposed a hybrid model that combines Temporal Convolutional Networks (TCN), Bidirectional GRU (BiGRU), and a self-attention mechanism for intrusion detection. TCN captures short-term patterns, BiGRU learns long-term dependencies, and attention highlights key features. Tested on the CSE-CIC-IDS2018 dataset, the model reached 97.83% accuracy and performed well on DDoS and botnet attacks, but had difficulty detecting infiltration attacks due to class imbalance.The studies in this section show that combining CNN and RNN models is effective for capturing both spatial and temporal features in vehicular network traffic.

2.4.3 Multi-RNN and Autoencoder-Based Models

Some studies explore advanced models that combine different RNN types or include unsupervised learning. For example, [84] proposed a hybrid LSTM-GRU model with a DENSE (ReLU) layer to build a fast and accurate intrusion detection system for both inter and intra-vehicle networks in IoV. By combining LSTM and GRU, the model improves learning and reduces detection time, targeting attacks like DDoS, spoofing, and fuzzing. It was tested on two datasets: a merged inter-vehicle DDoS dataset and the Car-Hacking CAN bus dataset. After preprocessing and balancing with SMOTE, it achieved excellent results 99.87% F1-score for DDoS and 99.99% for multi-class detection outperforming standalone RNNs and traditional machine learning models.

Another relevant work [85] proposed a hybrid model that combines an LSTM-based Variational AutoEncoder (LSTMVAE) for feature extraction with a BiGRU for classifying multiple types of attacks in IoV networks. The LSTMVAE compresses time-series data into a lower-dimensional form, while the BiGRU captures patterns in both directions to improve accuracy. A softmax layer handles the final classification. Tested on the ToN-IoT dataset, which includes various attack types, the model achieved 99.3% accuracy, with F1-score, precision, and recall between 94% and 100%.These studies show that combining different RNN types with autoencoders is highly effective for detecting complex and diverse attacks in vehicular networks. While most works combine RNNs with CNNs, some studies show that CNN-based models alone can also deliver high accuracy.

2.4.4 Pure CNN Architectures

Besides hybrid and RNN-based methods, some recent studies focus purely on convolutional architectures. For example, [86] proposed TCAN-IDS, a CNN-based intrusion detection system for IoV. The model uses a Temporal Convolutional Attention Network (TCAN), which combines spatial-temporal convolutions with a global attention mechanism to address the limits of traditional CNNs and RNNs. It was tested on a CAN bus dataset from the Hacking and Countermeasures Research Lab (HCRL), targeting attacks like DoS, Fuzzy, and Spoofing. The model achieved near-perfect F1-scores—99.98% for DoS, 99.92% for Fuzzy, and 99.97% for Spoofing—showing that convolutional models with attention can be very effective even without RNNs.

Overall, these studies confirm that deep learning offers powerful solutions for intrusion detection in IoV networks.

2.5 Summary Table

Tables 2.1 and 2.2 summarize the deep learning-based intrusion detection approaches reviewed in this chapter. They compare the approaches based on the models, datasets, types of attacks, relevance to IoV, accuracy and other metrics results.

Author & Year	**Model Used**	**Dataset**	**Attack Types**	**IoV-Specific**	**Accuracy**
Zhang et al. (2020)[77]	LSTM	SUMO + NS2 (simulated)	Sybil (identity-based)	Partially	95%
Kan et al. (2023)[78]	BiLSTM + Attention	HCRLab + Korean Public	Replay, DoS, Fuzzing	Yes	92.8%, 97.9%, 95.8%
Xing et al. (2023)[79]	TCN + LSTM + Attention + MLP (PA-STF)	NSL-KDD, UNSW-NB15	DoS, Probe, Exploits, R2L, U2R, Shellcode	No	98.68%, 96.34%
Wang et al. (2023)[80]	GCN + BiLSTM + Attention	NSL-KDD, CICIDS2017	DoS, DDoS, PortScan, Probe, R2L	No	95.87%, 94.12%
Manderna et al. (2023)[81]	Self-Attention + BiLSTM + CCNN	KDD-CUP99, ToN-IoT, VeReMi	DoS, DDoS, Binary (VeReMi)	Yes	99%, 99%, 98.6%
Javed et al. (2021)[82]	CNN + Attention-based GRU (CANintelliIDS)	Custom CAN (KIA Soul)	DoS, Fuzzy, Impersonation	Yes	94.06%, 95.09%, 94.01%
Ullah et al. (2022)[84]	LSTM + GRU + Dense (HDL-IDS)	CIC + CAN Bus	DDoS, Spoofing, Fuzzy	Partially	Up to 99.99%
Kanika Aggarwal (2023)[85]	LSTM-VAE + BiGRU + Softmax	ToN-IoT	DDoS, MITM, Scanning, Injection, XSS, Ransomware, Backdoor	No	99.3%
Song et al. (2023)[83]	TCN + BiGRU + Attention (TGA)	CSE-CIC-IDS2018	DDoS, DoS, Brute Force, Infiltration, Botnet, Web Attacks	No	97.83%
Cheng et al. (2022)[86]	TCN + Global Attention (TCAN-IDS)	Custom CAN (HCRL)	DoS, Fuzzy, Spoofing	Yes	F1: 99.98%

Table 2.1: Comparative Summary – Part 1: Accuracy of Deep Learning Models for IoV Intrusion Detection

Author & Year	Recall	F1-Score	FP Rate	FN Rate	Execution Time
Zhang et al. (2020)[77]	∼95%	Not reported	∼5%	∼5%	∼300s (simulation)
Kan et al. (2023)[78]	90.79%, 99.73%, 99.35%	Not reported	2.91%, 3.39%, 5.87%	∼9.21%, ∼0.27%, ∼0.65%	38.3s (total)
Xing et al. (2023)[79]	Not reported	98.94%, 95.76%	0.21%, 1.38%	∼1.32%, ∼3.66%	1.45ms/message
Wang et al. (2023)[80]	95.87%, 94.72%	94.25%, 94.36%	6.31%, 6.01%	Not reported	Not reported
Manderna et al. (2023)[81]	Not reported	Not reported	Not reported	Not reported	Not reported
Javed et al. (2021)[82]	94.22%, 95.31%, 94.25%	94.06%, 95.09%, 94.01%	Not reported	Not reported	Not reported
Ullah et al. (2022)[84]	99.83% (Nadam)	99.87% (Nadam)	Not reported	Not reported	692s (DDoS), 51s (CAN)
Kanika Aggarwal (2023)[85]	94.67–100% (per-class)	94.36–99.99% (per-class)	0.000007–0.002728 (per-class)	Not reported	Not reported
Song et al. (2023)[83]	97.83%	97.57%	Not reported	Not reported	Not reported
Cheng et al. (2022)[86]	Not reported	99.98%	Not reported	Not reported	Not reported

Table 2.2: Comparative Summary – Part 2: Recall, F1, and Runtime of Deep Learning Models for IoV Intrusion Detection

Although the recent deep learning-based intrusion detection systems (IDS) for the Internet of Vehicles (IoV) show strong results, several important limitations remain. Many models overlook identity-based Advanced Persistent Threats (APTs), which are especially relevant in IoV settings. A common issue is the use of generic datasets like NSL-KDD or CICIDS2017 that don't reflect real IoV scenarios such as CAN-bus traffic or vehicle-to-vehicle misbehavior. Some models rely only on CAN IDs or timing data and ignore important anomalies in the payload. While BiLSTM and GRU models are used to capture time-based patterns, they often struggle with stealthy or evolving attacks.

To address these limitations, we propose a hybrid deep learning model tailored for detecting identity-based APTs in IoV, using the VeReMi dataset. The model combines four key components: a Temporal Convolutional Network (TCN) for capturing short-term patterns, a Transformer Encoder for learning global dependencies, a Bidirectional GRU (BiGRU) for understanding sequences in both directions, and a Squeeze-and-Excitation (SE) block for adjusting feature importance.

Our architectural choices are based on insights from two key studies discussed in the state of the art. TGA [83] shows that combining TCN and BiGRU effectively captures both short-term

and sequential patterns in time-series intrusion detection, supporting our use of these components for IoV. TCAN-IDS [86] confirms that are well-suited for modeling temporal features in IoV data. Building on these foundations, we added two new components—a dual-branch fusion (parallel TCN + Transformer) and a Squeeze-and-Excitation (SE) block—to overcome the limitations identified in earlier works.

The dual-branch fusion processes input in parallel in order to capture both short- and long-range temporal patterns. This helps in detecting complex identity-based APTs by learning multi-scale behaviors. After the BiGRU layer, a Squeeze-and-Excitation (SE) block highlights the most important features. Even though these techniques have been used in other fields, their combined use in the IoV context with the VeReMi dataset is novel. Together, they form an effective model for detecting identity-based intrusions in IoV networks

2.6 Conslusion

This chapter reviewed recent deep learning approaches for intrusion detection in vehicular networks, focusing on the strengths of RNN-based architectures in modeling sequential data. We categorized related works into four groups: pure RNN models, hybrid CNN-RNN architectures, multi-RNN and autoencoder models, and pure CNN methods. While these approaches achieved high accuracy, we identified key limitations including the lack of focus on identity-based APTs and insufficient use of realistic vehicular datasets. These gaps motivate our proposed hybrid model, introduced in the next chapter, which combines temporal, spatial, and attention-based components for robust intrusion detection in IoV environments using the VeReMi dataset.

Chapter 3

Architecture of the Proposed Model, Validation, and Contributions

3.1 Introduction

In modern IoV environments, identity-based attacks such as Sybil threats can mislead vehicles into rerouting or causing artificial congestion—posing serious safety risks. These attacks are often components of Advanced Persistent Threats (APTs), which are stealthy, adaptive, and difficult to detect using conventional intrusion detection systems (IDS).

To address this challenge, this chapter presents an AI-driven approach designed to detect identity-based intrusions by learning temporal and behavioral patterns in vehicular communication data. The work focuses on developing and evaluating a hybrid deep learning model tailored to this task.

The chapter begins by describing the VeReMi dataset and the selected identity-based attack scenarios. It then outlines the preprocessing pipeline, feature engineering steps, and label preparation. The architecture of the proposed model, combining Temporal Convolutional Networks (TCN), Transformer Encoders, Bidirectional GRU (BiGRU), and Squeeze-and-Excitation (SE) blocks is presented, alongside benchmark models used for comparison.

Finally, training strategies, hyperparameter choices, and evaluation metrics are discussed to assess model performance in detecting identity-based APTs within the IoV context.

Key Contributions

- Proposed a hybrid deep learning model for detecting identity-based threats in IoV.
- Adapted and cleaned the VeReMi dataset to focus on identity-based attack scenarios.
- Applied temporal modeling techniques to improve detection of stealthy APT behavior.
- Evaluated the model against standard benchmarks using IDS performance metrics.

3.2 Development Environment

This section describes the tools and libraries used to implement and evaluate the proposed deep learning model:

- **Platform and Programming Language** :Model training and evaluation were performed using Google Colaboratory (Colab), a free cloud-based platform that allows users

to write and execute Python code in a Jupyter Notebook environment. It provides access to computational resources, including GPU acceleration, making it well-suited for training deep learning models. To ensure faster processing and efficient experimentation, we used Google Colab Pro with GPU support. Initial development and testing were conducted locally using Visual Studio Code[87].

Figure 3.1: Google Colaboatory

Python [88], a high-level and general-purpose programming language, is widely used in data science and machine learning due to its simplicity, rich ecosystem, and extensive library support. Colab integrates many of these popular libraries by default, offering a convenient and powerful setup for rapid prototyping and experimentation.

Figure 3.2: Python

- **Libraries and Frameworks:** This section presents the key libraries, frameworks, and functions used during the development, training, and evaluation of the model.

  1. **Data Processing:**
     - `Pandas:` Used for data manipulation, exploration, and cleaning through powerful DataFrame structures.
     - `NumPy:` Provides support for numerical operations and array manipulation.
  2. **Data Preprocessing:**
     - `StandardScaler` **and** `RobustScaler`(from `sklearn.preprocessing`:) Used for feature scaling to normalize data distributions and handle scaling robust to outliers.
     - `train_test_split` (from `sklearn.model_selection`): Splits the dataset into training and test sets in a reproducible way.
     - `IsolationForest:` Used to identify mislabeled data points.
  3. **Model Evaluation:**
     - `classification_report:` Generates precision, recall, F1-score, and support metrics for classification tasks.

- `confusion_matrix`: Provides a matrix to evaluate model performance in terms of true/false positives and negatives.
- `roc_auc_score`: Computes the Area Under the Receiver Operating Characteristic (ROC AUC), a key metric for binary classification.

4. **Visualization:**
   - `Matplotlib`: A 2D plotting library used to visualize model performance metrics, such as training history and confusion matrices.
   - `Seaborn`: Built on top of matplotlib, it enables more attractive and informative statistical plots, particularly for heatmaps and distribution graphs.
5. **Deep Learning Framework :** `Keras` (within `TensorFlow 2.x`)
   Core Layers Used:
   - `LSTM`: Long Short-Term Memory units for modeling temporal dependencies in sequential data.
   - `GRU`: Used in BiGRU, GRU-RCAL, and hybrid architectures.
   - `Bidirectional`: Wraps RNN layers like LSTM or GRU to process input in both forward and backward directions.
   - `Dense`: Fully connected layers for classification.
   - `Dropout`: A regularization technique to reduce overfitting by randomly deactivating neurons during training.
   - `Conv1D`, `ReLU`, `GlobalMaxPooling1D`, `GlobalAveragePooling1D`, `Multiply`, `Concatenate`, `Lambda`: Extensively used in CNN and hybrid models like TCN-Transformer.

   Utilities and Functions:
   - `Sequential`: A linear stack of layers used to build the model architecture.
   - `EarlyStopping`: A callback that stops training when validation performance no longer improves.
   - `ModelCheckpoint`: Used to save the best-performing model during training.
   - `model.fit`: Trains the model using the provided training data.
   - `model.evaluate` **and** `model.predict`: Used for assessing model performance and generating predictions on unseen data.
6. **Transformer Components:**
   - `MultiHeadAttention`, `LayerNormalization`, `Add`: Used in the TCN–Transformer–BiGRU models.
7. **Graph Neural Networks:**
   - `PyTorch + PyTorch Geometric`: Used in a GAT+BiLSTM hybrid model for graph-based detection.

3.3 Dataset Description

The Vehicular Reference Misbehavior (VeReMi) dataset [89, 90] is a publicly available simulated dataset developed for evaluating misbehavior detection mechanisms in Vehicular Ad-Hoc Networks (VANETs). It was generated using the LuST traffic scenario (v2) and a modified version of the VEINS simulator (v4.6) which models realistic urban vehicular movement and wireless communication.

VeReMi captures message-level data exchanged among vehicle on-board units (OBUs), focusing on Basic Basic Safety Message (BSM)s and GPS-based self-reports. Each message is labeled with ground truth indicating whether it is legitimate or malicious, making it well-suited for supervised learning in intrusion detection tasks.

The core structure consists of:

- Per-message logs including sender identity, timestamp, velocity, position, and other telemetry.
- Per-vehicle reception logs capturing what each vehicle receives and from whom—crucial for assessing trust and consistency.
- Simulations are run under three traffic densities (low, medium, high), five attack types (Constant, Constant Offset, Random, Random Offset, Eventual Stop), and three attacker densities (10%, 20%, 30%), enabling scalability in benchmarking detection models.

To better support AI-based intrusion detection and attack type differentiation, we used the VeReMi Extension Simple dataset [91] from Kaggle, which extends the original VeReMi dataset by:

- Adding explicit `Attack` and `Attack_type` fields for each message.
- Covering 18 labeled classes, including normal behavior, identity-based attacks (e.g., Replay, Sybil variants), and other forms of position manipulation.
- Providing 2.2 million messages (in `.csv` format), with approximately 1.9 million labeled as normal and 300,000 as attacks.

This enhanced labeling supports binary and multi-class classification, and includes identity-relevant metadata such as:

- `senderPseudo`: a pseudonym used in place of real vehicle IDs.
- `messageID` and `sender`: for tracking message-source relationships over time.

The dataset's structure and rich annotations make it well-suited for detecting identity-based Advanced Persistent Threats (APTs) in Internet of Vehicles (IoV) environments. While APTs are typically multi-stage and stealthy, several VeReMi attack types (e.g., DoS random Sybil, Traffic congestion Sybil, Data replay) mimic the tactics used in identity-based deception, allowing researchers to simulate and evaluate defensive models against identity misuse scenarios. Table 3.1 below provides an overview of the dataset's key features and their descriptions.

Feature	Description
type	Message type: 2 = GPS (self), 3 = BSM
sendTime	Timestamp of message transmission
sender	Unique identifier of the sending vehicle
senderPseudo	Pseudonym used instead of real vehicle ID
messageID	Unique identifier for the message packet
class	Integer code (0–16) indicating behavior or attack type
posx, posy, posz	3D spatial coordinates
spdx, spdy, spdz	Velocity components in x, y, z
aclx, acly, aclz	Acceleration components in x, y, z
hedx, hedy, hedz	Heading/orientation components
Attack	Categorical label: “normal”, “attack”, or “fault”
Attack_type	Text label describing specific attack (e.g., "Constant position", "Data replay")

Table 3.1: VeReMi Extension–Simple Dataset Feature Descriptions

3.4 Data Preprocessing

To make sure the VeReMi dataset was suitable for training deep learning models, a series of preprocessing steps were applied. These steps are detailed below and were essential for improving the data quality and the model performance.

1. **Step 1 – Dataset Filtering:**
   After analyzing the different attack types present in the VeReMi Extension Simple dataset, we performed a targeted selection of those attacks that reflect identity-based Advanced Persistent Threat (APT) characteristics in the context of the Internet of Vehicles (IoV). The selected attack types are: DoS random sybil, Traffic congestion sybil, Data replay sybil, DoS disruptive sybil, Data replay, Random position, and Constant position. A detailed overview of these attacks, including their definitions and APT-relevant traits, is provided in Table 3.2.

Full Name	Definition (Short)	APT-Relevant Traits
DoS Random Sybil	Injects numerous bogus vehicle identities at random to flood the network.	Identity spoofing, availability disruption
Traffic Congestion Sybil	Sends fake identities to simulate traffic congestion and mislead routing decisions.	Strategic deception, resource exhaustion
Data Replay Sybil	Replays valid messages using fake IDs to appear trustworthy.	Stealth, impersonation
DoS Disruptive Sybil	An aggressive form of DRS that disrupts the network by flooding it with high-frequency Sybil messages.	Flooding, persistent attack
Data Replay	Reuses previously captured legitimate messages to mimic real vehicle behavior.	Stealth, deception, evasion
Random Position	Sends fabricated and inconsistent location data.	Identity confusion, misleading trust logic
Constant Position	Continuously broadcasts a fixed position, masking real movement.	Obfuscation, stealth

Table 3.2: Summary of Identity-Based Attacks and Their APT-Relevant Traits

The selected attacks from the dataset were chosen for their manipulation of vehicular identity or message authenticity, aligning them with identity-based threats. Sybil attacks simulate multiple fake identities, replay attacks reuse legitimate messages to mimic real behavior, and position falsification misleads others using false location data. These tactics reflect APT-like traits—being advanced, persistent, and targeting trust or availability. While VeReMi does not explicitly model APTs, these attacks exhibit identity-based APT behavior within the IoV context, justifying their inclusion in this study

2. **Step 2 — Feature Elimination:**
   To enhance the learning quality of the model and prevent overfitting or information leakage, several columns were excluded from the dataset after careful inspection. Each feature was assessed based on its relevance, consistency, and potential to bias the model.

   Type and class were dropped as they serve internal simulation purposes only. type provides general node categories (e.g., car, RSU), while class defines simulation-specific roles or behaviors. Neither is used in real-time detection nor available in real deployments, and keeping them could introduce bias or unrealistic assumptions.

   Sender, senderPseudo, and messageID were excluded to prevent identity-based leakage. These fields risk allowing the model to memorize specific nodes instead of learning behavioral patterns. SenderPseudo revealed 15,140 unique pseudonyms with an attack ratio of 1.0, meaning they appeared only in attack samples and never in normal ones. This is a strong indicator of identity leakage, where the model could incorrectly associate a sender ID with maliciousness—breaking generalizability in real-world IDS deployment.

   Posz, spdz, aclz, and hedz (Z-axis motion and orientation data) were consistently null, constant, or unused due to VeReMi's 2D simulation nature and thus provided no discriminative power.

These removals were not arbitrary; they were guided by:

- Careful review of dataset documentation and simulation design
- Empirical inspection (e.g., identifying constant or missing-value-heavy features)
- Best practices for avoiding feature leakage and ensuring real-world applicability

A correlation matrix was not used for this initial feature elimination step because the goal was not to reduce redundancy among numerical features, but to remove structurally unfit or semantically misleading ones.

3. **Step 3 — Label Cleaning and Correction:**
In the original dataset, some samples were labeled as Fault, which do not explicitly fall under conventional "attack" or "normal" categories. However, in the context of intrusion detection systems (IDS) for vehicular networks, fault behaviors often indicate anomalous or system-compromising actions, such as disrupted communication or misbehaving nodes. Therefore, all fault labels were reclassified as attack to align with the binary intrusion detection goal.

   To further refine the correctness of the binary labels (attack = 1, normal = 0), we used a conservative relabeling strategy based on anomaly detection: An Isolation Forest was trained using only the data labeled as normal to learn the typical behavior pattern. It then assigned an anomaly score to each sample—lower scores indicating higher deviation from normal behavior. Based on these scores, samples originally labeled as normal but scoring in the lowest 5% (i.e., high anomaly) were relabeled as attack. Conversely, samples labeled as attack but scoring as highly normal (i.e., above that 5% threshold) were relabeled as normal.
   Although Attack_type contains detailed class names, it was not used for labeling because binary classification is a practical choice for intrusion detection systems (IDS) in vehicular networks, where rapid, decisive responses are critical. IDS must quickly determine whether behavior is malicious or benign—detailed attack type classification is secondary to ensuring timely threat mitigation.

4. **Step 4 — Feature Engineering:**
To improve detection performance, we engineered new features that capture temporal and behavioral dynamics in vehicle movements. These features were derived using spatial and temporal relationships between motion attributes such as velocity, acceleration, direction, and time. The goal was to extract behavioral signals that are harder to forge in identity-based APT attacks. Table 3.3 presents the derived features with their calculations and interpretations.

   To compute these features, the dataset was sorted chronologically using 'sendTime'. Differences were then calculated between each record and the one immediately preceding it. Although the dataset lacks explicit sender IDs due to anonymization, sorting by 'sendTime' provides a practical temporal sequence that captures useful behavioral transitions.

New Feature	How it was calculated	Meaning/Interpretation
speed	sqrt(spdx^2 + spdy^2)	Total speed of the vehicle based on its velocity vector
acceleration	sqrt(aclx^2 + acly^2)	Overall acceleration derived from X and Y components
direction	arctan2(spdy, spdx)	Direction angle of the velocity vector (in radians)
delta_speed	Difference between current and previous speed value	Captures change in vehicle speed
delta_acceleration	Difference between current and previous acceleration value	Detects sudden or subtle acceleration shifts
delta_direction	Difference between current and previous direction value	Measures change in motion direction
time_gap_to_prev _message	Difference between current and previous sendTime	Time interval between consecutive messages

Table 3.3: Derived features with their calculations and interpretations.

These delta-based features provide temporal signatures that attackers typically fail to replicate accurately. While static values like position or speed may be spoofed, consistent and realistic variations in motion (like acceleration shifts or direction changes) are difficult to fake. APT attacks often fail to replicate natural vehicle physics. For instance:

- **Sybil Attacks (e.g., DoS disruptive sybil, Traffic congestion sybil)**: Betrayed by inconsistencies in delta_direction and delta_speed across fake identities, as coordinated malicious nodes struggle to maintain physically plausible movements.
- **Random Position Attacks (non-Sybil position spoofing):** Exposed through erratic direction changes and impossible speed/acceleration pairs (e.g., sudden 90° turns at high speeds).
- **Replay Attacks (e.g., Data replay sybil):** Detected via identical delta_acceleration sequences and near-zero time_gap_to_prev_message values, unlike natural vehicle behavior.

These features are not only observable in real-world vehicular networks equipped with GPS systems, but are also computationally lightweight, making them suitable for deployment in real-time IDS systems. The inclusion of temporal dynamics (via delta and timing features) allows the model to capture subtle anomalies and deviations in identity-based APT behavior that would otherwise be missed in static data.

5. **Additional Data Cleaning :**

   - **Handled missing values:** Rows containing NaN values (especially after delta feature generation) were removed using ‘dropna()‘ to ensure data integrity.
   - **Removed duplicate rows:** Duplicate entries were dropped to prevent redundancy and potential model overfitting.
   - **Fixed data types:** Fields originally encoded as objects (e.g., due to parsing errors) were converted to appropriate numeric types where possible.

- **Removed erroneous or malformed data:** Any rows with infinite values or corrupted entries (such as those caused by simulation glitches or scientific notation issues) were cleaned using replacement and filtering methods.
- **Applied robust feature scaling:** Numerical features were scaled using 'RobustScaler', which centers features using the median and scales according to the interquartile range. This approach is resilient to outliers and prepares the data effectively for deep learning.
- **Performed balanced extraction:** From the cleaned and scaled dataset, an equal number of attack and normal samples were selected to create a balanced dataset of approximately 384,546 rows. This ensures fair and unbiased model training.

3.5 Model Architecture

The proposed model aims to detect identity-based Advanced Persistent Threats (APTs) in vehicular networks by capturing both short and long-term behavioral patterns. These threats—such as Sybil, replay, or falsified identity attacks exploit the identity layer in Internet of Vehicles (IoV) environments by imitating legitimate vehicles over time.

To counter such threats, the model adopts a hybrid deep learning architecture that combines four key components: **Temporal Convolutional Networks (TCNs)**, **Transformer Encoders**, **Bidirectional Gated Recurrent Units (BiGRUs)**, and a **Squeeze-and-Excitation (SE) block**. Each of these components plays a distinct role in capturing features associated with identity manipulation, as summarized in Table 3.4.

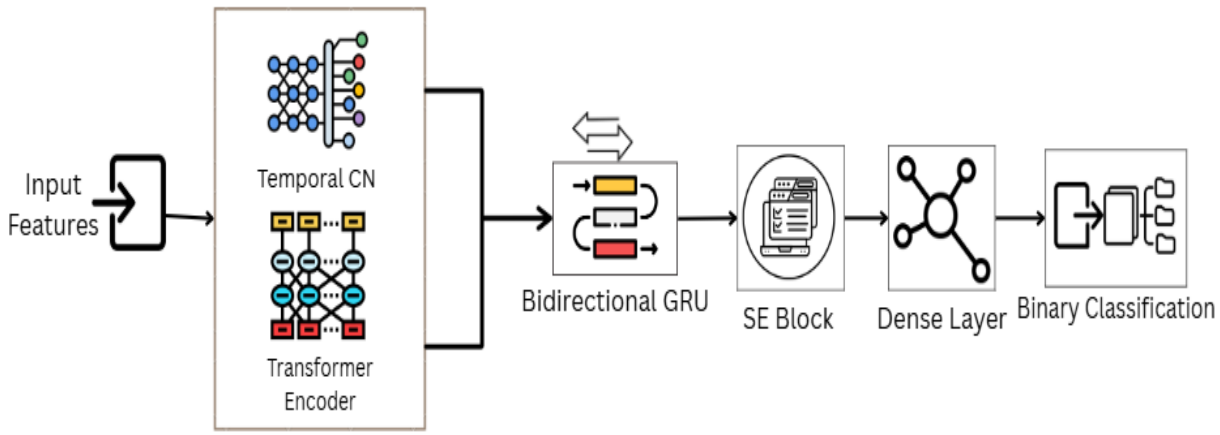

Figure 3.3: Proposed Architecture for Identity-Based APT Detection in IoV.

Figure 3.3 illustrates the end-to-end structure of the proposed model. Each input sample is formatted as a sequence of feature vectors representing vehicular behavior over time, including attributes such as speed, position, acceleration, and direction. This format enables the model to learn temporal and contextual dependencies without relying on large time windows.

The architecture operates in sequential stages. First, temporal and global features are extracted through the TCN and Transformer branches, respectively. Their outputs are fused and passed to a BiGRU layer for temporal modeling. A Squeeze-and-Excitation block then refines the learned representation by emphasizing the most relevant features. Finally, a dense sigmoid-activated layer produces a binary output indicating whether the input sequence corresponds to an attack or normal behavior.

Module	**Functionality for Identity-Based APT Detection**
Temporal Convolution (TCN)	Captures fine-grained temporal variations in movement patterns (e.g., abrupt speed changes or erratic direction shifts), which can indicate spoofed behavior.
Transformer Encoder	Learns global correlations among features (e.g., multiple vehicles with similar fake IDs or movement), enabling detection of coordinated or clustered identity attacks.
Bidirectional GRU (BiGRU)	Models identity evolution over time, detecting gradual mimicry or replay behavior patterns in both forward and backward temporal directions.
Squeeze-and-Excitation (SE) Block	Applies channel-wise attention to dynamically adjust the importance of identity-relevant features while suppressing noise.

Table 3.4: Functional Contributions of Model Components in Identity-Based APT Detection.

To complement the overview, Table 3.5 presents the configuration of each component in the architecture, including input formatting, layer parameters, and activation functions.

Component	**Description**
Input Shape	`(time_steps, num_features)`
Temporal Convolution	1D causal Conv1D with 64 filters and kernel size = 1, using ReLU activation to extract local temporal patterns.
Transformer Encoder	Multi-head self-attention (2 heads), followed by residual connections and layer normalization to model global dependencies.
Fusion Layer	Concatenation of the TCN and Transformer outputs, combining local and global features.
Recurrent Layer	BiGRU with 64 units and return sequences enabled, for bidirectional temporal modeling of identity behavior.
Squeeze-and-Excitation Block	Channel-wise attention block with reduction ratio = 8, to adjust feature importance dynamically.
Output Layer	Dense(1) with sigmoid activation for binary classification: `attack = 1`, `normal = 0`.

Table 3.5: Component Summary of Proposed Model Architecture.

3.6 Training and Configuration

The proposed model was trained using a specific configuration. The main hyperparameters and training strategies are summarized in Table 3.6.

Aspect	**Details**
Loss Function	Binary Cross-Entropy (BCE)
Optimizer	Adam optimizer with a learning rate of 0.001
Epochs	Trained for up to **30 epochs**, with **early stopping** enabled (patience = 10)
Validation Strategy	20% of the training data is set aside for validation
Regularization	Dropout (rate = 0.3) is applied after dense layers to reduce overfitting
Batch Size	128 — chosen to balance training efficiency and generalization
Feature Scaling	RobustScaler — a median-centered, outlier-tolerant scaling method

Table 3.6: Training Configuration and Hyperparameters of the Proposed Model.

This architecture is particularly suited for real-time IDS deployment in vehicular networks where identity-based threats must be detected early. The combination of sequence-aware, attention-based, and feature enhancing components makes it both sensitive to subtle identity anomalies and resilient against evasion.

3.7 Experimental Results

This section presents and analyzes the results of the proposed model when trained and tested on the cleaned VeReMi dataset. It begins with the definitions and formulas of the evaluation metrics used, followed by a performance summary, detailed analysis, and visual interpretation of the outcomes.

3.7.1 Evaluation Metrics

The effectiveness of the model is assessed using standard classification metrics: Accuracy, False Positive Rate (FP Rate), False Negative Rate (FN Rate), ROC AUC, and Mean Error Rate (MER).

- **True Positive Rate (TP Rate)**—also known as recall or sensitivity—measures the proportion of actual attack samples correctly classified as attacks. It is calculated as:

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

- **True Negative Rate (TN Rate)**—also known as specificity—indicates the proportion of normal samples correctly classified as normal. It is given by the formula:

$$\text{TN Rate} = \frac{TN}{TN + FP}$$

- **False Positive Rate (FP Rate)** indicates the proportion of normal samples incorrectly predicted as attacks. It is computed using:

$$\text{FP Rate} = \frac{FP}{FP + TN}$$

- **False Negative Rate (FN Rate)** measures the proportion of attack samples wrongly classified as normal. It is defined as:

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

- **Accuracy** measures the overall proportion of correct predictions, both true positives and true negatives, among all predictions. It is calculated by:

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

- **Roc! (Roc!)-Area Under the Curve (AUC)** reflects the model's ability to distinguish between the two classes across all classification thresholds. A value closer to 1 indicates superior performance.

- **Mean Error Rate (MER)** is the average of the false positive and false negative rates, providing a balanced indication of misclassification errors. The formula is:

$$\text{MER} = \frac{\text{FP Rate} + \text{FN Rate}}{2}$$

- **Confusion Matrix** is a tabular representation used to evaluate the performance of a classification model by showing the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It helps visualize how well the model distinguishes between classes by comparing predicted labels to actual labels.

3.7.2 Model Performance

The proposed model—combining Temporal Convolution (TCN), Transformer Encoder, Bidirectional GRU (BiGRU), and a Squeeze-and-Excitation (SE) block—achieved the following results on the test set:

- **Accuracy:** 98.0%
- **False Positive Rate:** 3.5%
- **False Negative Rate:** 0.9%
- **ROC AUC:** 0.9973
- **Mean Error Rate:** 2.20%

These values are derived from the model's predictions on the test subset after final training convergence. The results demonstrate the model's ability to accurately identify identity-based APT attacks with minimal misclassification.

3.7.3 Result Analysis and Visualizations

The low FN rate is particularly critical for intrusion detection in IoV environments, where undetected threats may compromise vehicular safety. A moderate FP rate (3.5%) means the system avoids over-alerting for benign behavior, which is also essential for operational reliability.

The high ROC AUC score (0.9973) confirms the classifier's strong decision boundary and generalization capacity. This robustness stems from the complementary strengths of the architectural components—TCN (for capturing short-term dynamics), Transformer Encoder (for modeling global identity context), BiGRU (for learning sequential dependencies), and the SE block (for feature recalibration).

To support the numerical evaluation, the following figures illustrate the model's behavior graphically:

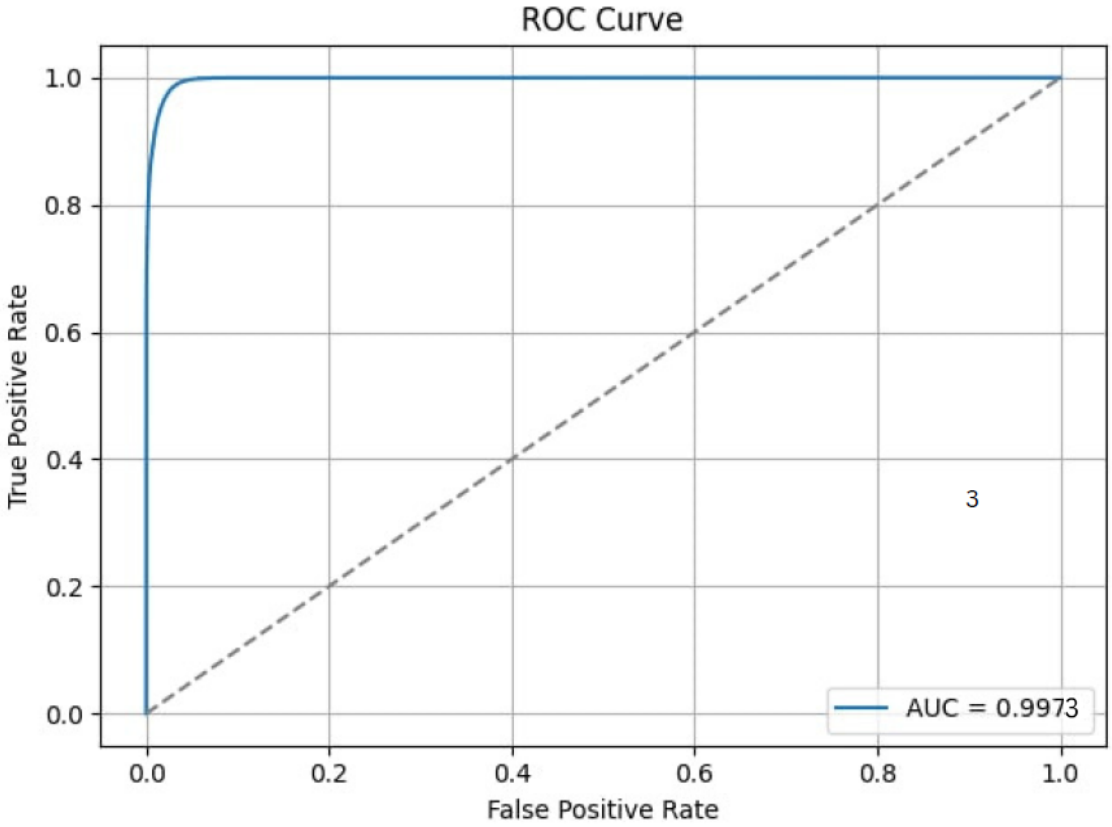

Figure 3.4: ROC Curve of the Proposed Model

Figure 3.4 shows the ROC curve, confirming excellent discrimination between the attack and normal classes across all thresholds.

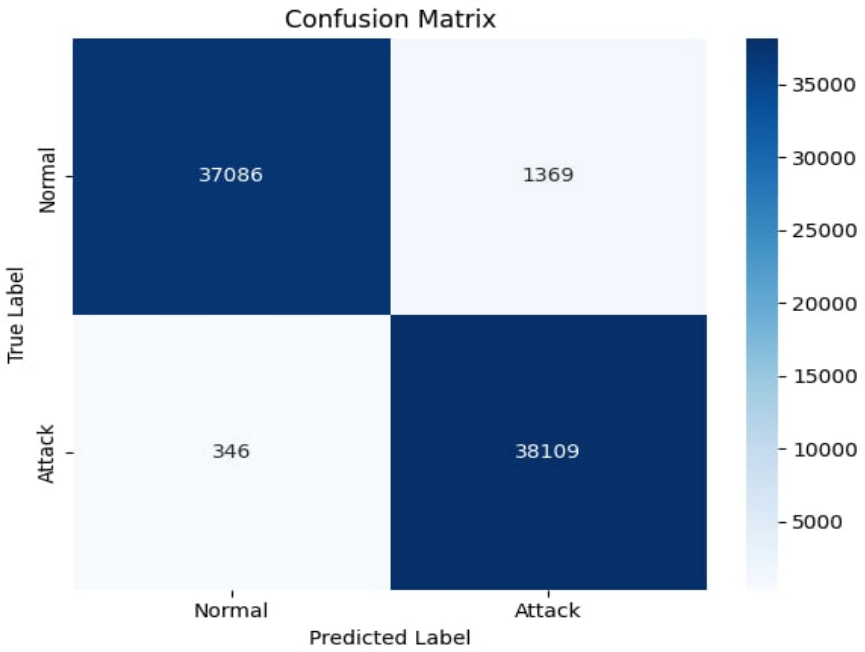

Figure 3.5: Confusion Matrix of Predictions

Figure 3.5 displays the confusion matrix. The majority of predictions fall in the true positive and true negative categories, indicating the model's effectiveness in correctly classifying both benign and malicious instances.

3.7.4 Cross-Validation Evaluation

In addition to evaluating the model on a traditional train/test split, 5-fold stratified cross-validation was performed to assess the stability and generalizability of the proposed architecture. This approach divides the dataset into five equal parts, using four for training and one for testing in each iteration, with class distribution preserved across folds.

Cross-validation helps reduce the risk of overfitting to a single data split and provides a more reliable estimate of the model's real-world performance. In the context of intrusion detection for IoV systems, it ensures that the model consistently detects identity-based APTs across varying traffic patterns and scenarios.

The results were highly consistent across folds, with the model achieving a mean accuracy of 97.73% and a mean ROC AUC of 0.9973, both with minimal standard deviation. These values demonstrate excellent and reliable performance across varying data splits, confirming that the model is not overly dependent on a specific partition of the data.

Fold	Accuracy	ROC AUC
1	97.67%	0.9971
2	97.73%	0.9972
3	97.76%	0.9973
4	97.81%	0.9976
5	97.65%	0.9972
Mean	**97.73%**	**0.9973**
Std Dev	± 0.06%	± 0.0002

Table 3.7: 5-Fold Cross-Validation Results

The consistency across all folds indicates that the model is both robust and generalizable, with minimal variance in classification performance. Furthermore, balanced precision and recall scores across each fold show that the model maintains effective detection rates for both normal and attack classes.

3.8 Comparative Analysis

To assess the effectiveness of the proposed architecture, its performance is compared against a range of classical machine learning and deep learning models.

Model	Accuracy (%)	ROC AUC	FP Rate (%)	FN Rate (%)	Mean Error Rate (%)
Random Forest	96.0	0.9920	5.3	3.0	4.15
GRU	97.6	0.9968	3.4	1.6	2.50
CNN + BiGRU	98.0	0.9972	3.0	1.5	2.25
Stacked BiLSTM	98.0	0.9971	3.2	1.3	2.25
Self-Attn BiLSTM + Cascaded CNN	98.0	0.9972	3.3	1.2	2.27
Temporal CNN + LSTM + Attention + MLP	98.0	0.9972	3.4	1.2	2.30
GAT + BiLSTM	82.0	0.9215	11.4	22.4	16.90
Proposed (TCN + Transf. + BiGRU + SE)	98.0	0.9973	3.5	0.9	2.20

Table 3.8: Performance Comparison of Models on the VeReMi Dataset.

As seen in Table 3.8, most deep learning models achieved strong performance, especially those incorporating temporal sequence modeling (e.g., GRU, BiLSTM) and attention mechanisms. Architectures like CNN–BiGRU, Self-Attention BiLSTM + CNN, and Stacked BiLSTM achieved a strong balance between accuracy, low FP/FN rates, and high ROC AUC, making them highly suitable for detecting evolving identity-based APT behaviors in IoV environments.

In comparison, the classical machine learning model Random Forest also performed reasonably well, largely due to the quality of the input features. However, it relies on static, snapshot-based decisions and lack the temporal learning depth offered by RNNs and attention-based architectures.

The GAT + BiLSTM model performed significantly worse than others, with an FN rate above 22%, suggesting it was unable to extract meaningful patterns. This underperformance is likely due to the fact that the graph-based assumptions of GAT do not align well with the sequential and feature-centric nature of the VeReMi dataset. Unlike natural graphs (e.g., social networks), the synthetic graph sequences created from temporal slices here might lack structural richness, causing poor node embedding quality.

Notably, the overall closeness in performance (especially among the DL models) is not suspicious but rather expected. All models are trained on the same clean, well-structured dataset with strong spatio-temporal patterns. Still, each model offers nuanced trade-offs:

- Machine Learning Models : simple, interpretable, fast.
- Basic RNNs (GRU): efficient at temporal pattern learning.
- Hybrid Architectures (e.g., CNN+BiGRU, LSTM+Attention): strongest generalization across attack patterns

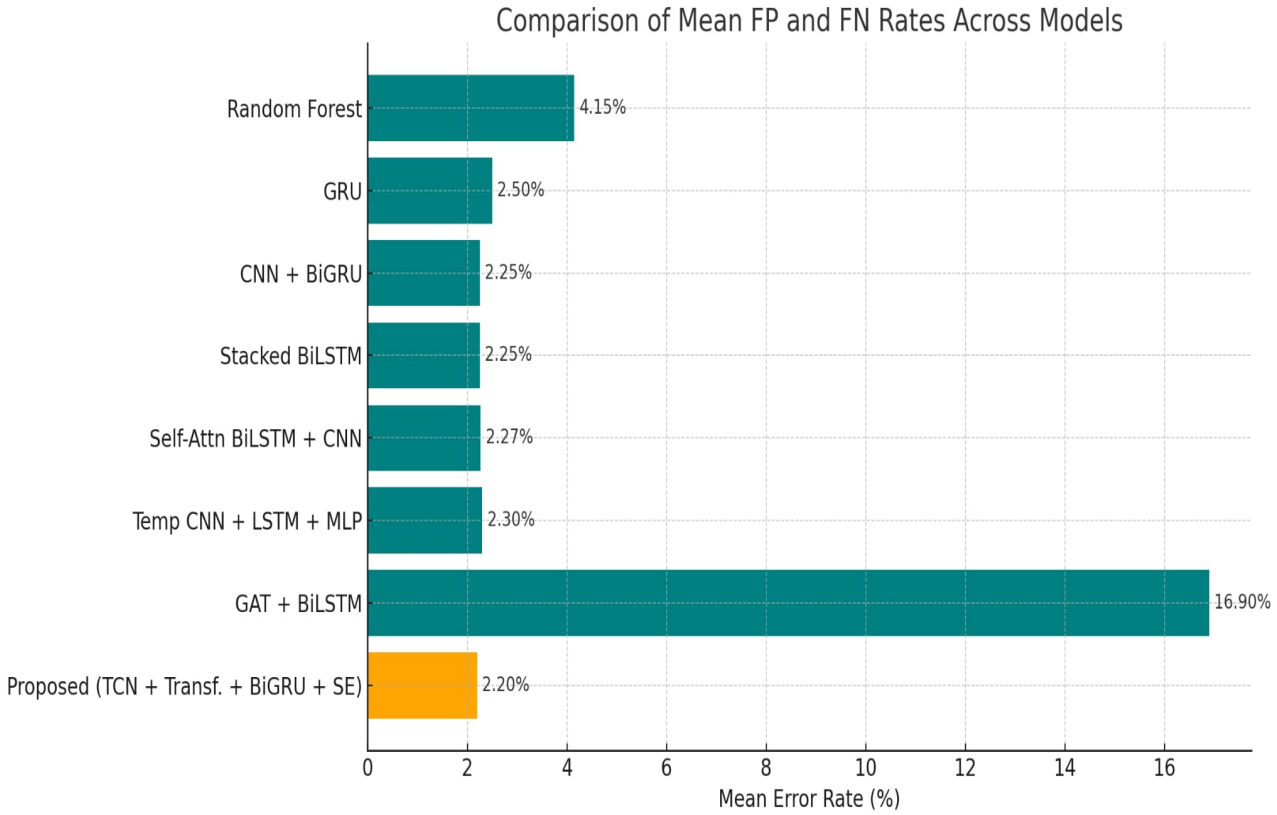

Figure 3.6: Comparison of Mean FN and FP Rates Across Models

Figure 3.6 presents a bar chart comparing the average false negative and false positive rates across multiple models. The proposed architecture clearly outperforms others in minimizing FN rate while maintaining a competitive FP rate, which is crucial for effective identity-based APT detection in vehicular networks.

3.9 Training Time and Practical Viability

As shown in Table 3.8, several high-performing models including BiLSTM, CNN+BiGRU, and Temporal CNN+LSTM+Attention+MLP achieved comparable detection accuracy (98%) and

low false positive rates ($\leq$3.5%) to our proposed TCN-Transformer-BiGRU-SE model. This performance consistency allows us to isolate training time as the key differentiating factor in evaluating real-world viability. The table below compares the training time and relative execution speed of various deep learning models evaluated during this study.

Model	**Training Time (s)**	**Relative Speed**
TCN-Transformer-BiGRU-SE (Proposed)	262.87	1.0× (Ref)
BiLSTM	518.90	1.97× slower
CNN+BiGRU	557.62	2.12× slower
Temporal CNN+LSTM+Attention+MLP	1149.29	4.37× slower

Table 3.9: Training Time Comparison of Different Models

Three architectural synergies, already detailed in Section 3.4, explain this efficiency:

- **Parallel Feature Extraction:** The TCN (local patterns) and Transformer (global context) process data concurrently, unlike sequential models like BiLSTM.
- **Lightweight Attention:** The SE block's channel-wise reweighting (reduction ratio=8) minimizes redundant computations.
- **BiGRU Consolidation:** Unlike stacked RNNs that rely on iterative memory gates, the single BiGRU layer captures temporal dependencies more efficiently without requiring multiple consolidation layers.

These findings challenge the common assumption that hybrid models must trade speed for accuracy, directly addressing Chapter 1's emphasis on real-time deployability. While future work should include on-device inference profiling, these training results already validate the proposed model's suitability for edge-based IoV systems.

3.10 Conclusion

This chapter detailed the development and evaluation of our proposed intrusion detection approach for identity-based APT-like threats in IoV. The experimental results obtained were very promising, surpassing the performance of existing approaches on the VeReMi Extension–Simple dataset and confirming the effectiveness of our architecture. These findings highlight the potential of AI-driven models in addressing complex identity-related threats, while also serving as a foundational step toward real-world IDS deployment.

General Conclusion

In this thesis, we explored the application of artificial intelligence techniques for detecting identity-based Advanced Persistent Threats (APTs) in the Internet of Vehicles (IoV). After presenting the foundational concepts related to IoV, identity-based threats, APTs, and Intrusion Detection Systems (IDS), we focused on the use of deep learning—particularly recurrent neural networks for modeling temporal behaviors and enhancing threat detection.

The state of the art review allowed us to categorize existing approaches into pure RNN-based models, hybrid CNN-RNN architectures, and multi RNN or autoencoder based frameworks. This analysis revealed the growing interest in deep learning for IoV security, while highlighting the need for more targeted solutions for identity-related attacks.

Our contribution centered on the design of a hybrid deep learning model tailored for identity-based APT detection. The proposed architecture integrates a Temporal Convolutional Network (TCN), a Transformer encoder, a Bidirectional GRU, a Squeeze-and-Excitation (SE) block, and dense layers. It was trained and evaluated using the VeReMi dataset, which we reinterpreted to align with identity misuse scenarios.

Significant effort was dedicated to data preprocessing, feature extraction, and experimental tuning in order to model behaviors such as Sybil and replay attacks as indicators of identity-based APT-like activity. The experimental results were promising and demonstrated the model's effectiveness in identifying subtle and persistent threats in vehicular networks.

While we developed a novel AI framework for detecting identity-based APT-like threats in IoV networks using simulated data to establish a baseline for behavioral anomaly detection, reliance on datasets like VeReMi serves primarily as a foundation for early-stage experimentation. Future work should focus on enhancing practical applicability by evaluating the model against real-world vehicular communication data. The binary classification approach adopted here marks an important initial step toward deployable intrusion detection, prioritizing reliable threat identification over granular attack categorization. Nonetheless, extending the framework to support multi-class classification—such as differentiating Sybil from Replay attacks—remains a promising direction for future research.

The proposed architecture's success in catching identity-based APT-like behavior may also open new research pathways, as the same temporal analysis principles could be adapted to detect more sophisticated, multi-stage APTs in IoV environments.

In summary, this work positions the model as a core component for future real-time IDS implementations, where its design could be integrated with vehicular edge computing infrastructure.

Bibliography

[1] Samira Yessad, Smail Hamadache, et al. "Application-Aware Opportunistic Routing Protocol for Traffic Violations Notification in Internet of Vehicles". In: *2022 5th International Symposium on Informatics and its Applications (ISIA)*. IEEE. 2022, pp. 1–6.

[2] Yani-Athmane Bennai, Samira Yessad, et al. "A flexible and adaptive medium access control protocol for improving quality of service in vehicular ad-hoc networks". In: *International Journal of Computers and Applications* 44.10 (2022), pp. 929–938.

[3] Yani-Athmane Bennai, Samira Yessad, et al. "Link Stability Based Routing Protocol for Highway Scenarios in Vehicular Networks". In: *2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)*. Vol. 6. IEEE. 2021, pp. 1–5.

[4] Samia Moulai Hacene, Samira Yessad, et al. "Link Quality Estimation for Reliable Data Dissemination in Vehicular Ad hoc Networks". In: *2021 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*. IEEE. 2021, pp. 1–6.

[5] Sedeng Danba, Jingjing Bao, et al. "Toward collaborative intelligence in IoV systems: Recent advances and open issues". In: *Sensors* 22.18 (2022), p. 6995.

[6] Arif Hakimi, Kamaludin Mohamad Yusof, et al. "A survey on internet of vehicle (iov): A pplications & comparison of vanets, iov and sdn-iov". In: *ELEKTRIKA-Journal of Electrical Engineering* 20.3 (2021), pp. 26–31.

[7] Rim Gasmi and Makhlouf Aliouat. "Vehicular ad hoc networks versus internet of vehicles-a comparative view". In: *2019 international conference on networking and advanced systems (ICNAS)*. IEEE. 2019, pp. 1–6.

[8] Djihane Benderradji and Isra Bouanane. "Une architecture IoV hétérogène pour le transfert des données dans la communication Véhicule-Infrastructure". Mémoire de master. Université de Kasdi Merbah Ouargla, 2022.

[9] Nadjet Azzaoui, Mohammed Azim Krima, et al. "Système de Détection d'Intrusion dans Les Réseaux D'Internet des Véhicules". Mémoire de fin d'étude. Université de Kasdi Merbah Ouargla, 2023.

[10] Fangchun Yang, Jinglin Li, et al. "Architecture and key technologies for Internet of Vehicles: a survey". In: *Journal of communications and information networks* 2.2 (2017), pp. 1–17.

[11] Juan Contreras-Castillo, Sherali Zeadally, et al. "Internet of vehicles: architecture, protocols, and security". In: *IEEE internet of things Journal* 5.5 (2017), pp. 3701–3709.

[12] Easa Alalwany and Imad Mahgoub. "Security and trust management in the internet of vehicles (IoV): Challenges and machine learning solutions". In: *Sensors* 24.2 (2024), p. 368.

[13] Maxim Raya and Jean-Pierre Hubaux. "The security of vehicular ad hoc networks". In: *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*. 2005, pp. 11–21.

[14] Bhuiyan Mustafa Tawheed. "Speed Offset Attack Detection in Vehicular Ad-Hoc Networks (VANETs) Using Machine Learning". Master's thesis. University of Windsor (Canada), 2022.

[15] Dott Lorenzo Ghiro and Marco Franceschini. *A neural-network based anomaly detection system and a safety protocol to protect vehicular network*. 2024.

[16] Nicholas Jaton. "Distributed Neural Network Based Architecture for DDoS Detection in Vehicular Communication Systems". Master's Thesis. University of Nebraska - Lincoln, 2021.

[17] Abdulaziz A Alsulami, Qasem Abu Al-Haija, et al. "Symmetrical simulation scheme for anomaly detection in autonomous vehicles based on LSTM model". Doctoral dissertation. 2022, p. 1450.

[18] Aawista Chaudhry. "A Framework for Modeling Advanced Persistent Threats in Intelligent Transportation Systems". Master's thesis. Queen's University (Canada), 2021.

[19] Euclides Carlos Pinto Neto, Hamideh Taslimasa, et al. "CICIoV2024: Advancing realistic IDS approaches against DoS and spoofing attack in IoV CAN bus". In: *Internet of Things* 26 (2024), p. 101209.

[20] Sulaiman M Karim, Adib Habbal, et al. "Architecture, protocols, and security in IoV: Taxonomy, analysis, challenges, and solutions". In: *Security and Communication Networks* 2022.1 (2022), p. 1131479.

[21] Panagiotis Papadimitratos, Levente Buttyan, et al. "Secure vehicular communication systems: design and architecture". In: *IEEE Communications magazine* 46.11 (2008), pp. 100–109.

[22] Nishant Sharma, Naveen Chauhan, et al. "Security challenges in Internet of Vehicles (IoV) environment". In: *First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE. 2018, pp. 203–207.

[23] Changguang Wang, Zimeng Dai, et al. "A Novel Identity-based Authentication Scheme for IoV Security." In: *Int. J. Netw. Secur.* 22.4 (2020), pp. 627–637.

[24] Keturahlee Coulibaly. "An overview of intrusion detection and prevention systems". In: *arXiv preprint arXiv:2004.08967* (2020).

[25] Noor Hazlina Abdul Mutalib, Aznul Qalid Md Sabri, et al. "Explainable deep learning approach for advanced persistent threats (APTs) detection in cybersecurity: a review". In: *Artificial Intelligence Review* 57.11 (2024), p. 297.

[26] Ping Chen, Lieven Desmet, et al. "A study on advanced persistent threats". In: *Communications and Multimedia Security: 15th IFIP TC 6/TC 11 International Conference, CMS 2014, Aveiro, Portugal, September 25-26, 2014. Proceedings 15*. Springer. 2014, pp. 63–72.

[27] R. S. Ross. *Managing Information Security Risk: Organization, Mission, and Information System View*. 2011.

[28] Talal Halabi, Omar Abdel Wahab, et al. "Protecting the internet of vehicles against advanced persistent threats: A Bayesian Stackelberg game". In: *IEEE Transactions on Reliability* 70.3 (2021), pp. 970–985.

[29] Adel Alshamrani, Sowmya Myneni, et al. "A survey on advanced persistent threats: Techniques, solutions, challenges, and research opportunities". In: *IEEE Communications Surveys & Tutorials* 21.2 (2019), pp. 1851–1877.

[30] Rakshanda Agarwal, Sai Satya Pranay, et al. "Identity-based security scheme in internet of vehicles". In: *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018, Volume 1*. Springer. 2019, pp. 515–523.

[31] Sohail Abbas, Manar Abu Talib, et al. "Blockchain-based authentication in internet of vehicles: A survey". In: *Sensors* 21.23 (2021), p. 7927.

[32] Sarah Ali Siddiqui, Adnan Mahmood, et al. "A survey of trust management in the internet of vehicles". In: *Electronics* 10.18 (2021), p. 2223.

[33] Andy Greenberg. *Hackers Remotely Kill a Jeep on the Highway—With Me in It*. Accessed: 2025-07-06. July 21, 2015. URL: `https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/`.

[34] Sherin Kalli Valappil, Lars Vogel, et al. "Advanced IDPS Architecture for Connected and Autonomous Vehicles". In: *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE. 2024, pp. 1779–1785.

[35] Asma Alfardus and Danda B Rawat. "Machine learning-based anomaly detection for securing in-vehicle networks". In: *Electronics* 13.10 (2024), p. 1962.

[36] Karen Scarfone, Peter Mell, et al. "Guide to intrusion detection and prevention systems (idps)". In: *NIST special publication* 800.2007 (2007), p. 94.

[37] Meriem Houmer, Mariya Ouaissa, et al. "Applying machine learning algorithms to improve intrusion detection system in IoV". In: *Artificial Intelligence of Things in Smart Environments: Applications in Transportation and Logistics* (2022), p. 35.

[38] Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges". In: *computers & security* 28.1-2 (2009), pp. 18–28.

[39] Kamran Siddique, Zahid Akhtar, et al. "Developing an intrusion detection framework for high-speed big data networks: A comprehensive approach". In: *KSII Transactions on Internet and Information Systems (TIIS)* 12.8 (2018), pp. 4021–4037.

[40] Chen Feng, Md Nazmus Uddin, et al. "JiNao - Scalable Intrusion Detection for Network Infrastructure". In: *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2017, pp. 1–7.

[41] Khisamova Zarina I, Begishev Ildar R, et al. "Artificial Intelligence and Problems of Ensuring Cyber Security." In: *International Journal of Cyber Criminology* 13.2 (2019).

[42] Muhammad Shoaib Akhtar and Tao Feng. "An overview of the applications of Artificial Intelligence in Cybersecurity." In: *EAI endorsed transactions on creative technologies* 8.29 (2021).

[43] Muhammad Ismaeel Khan, Aftab Arif, et al. "The Most Recent Advances and Uses of AI in Cybersecurity". In: *BULLET: Jurnal Multidisiplin Ilmu* 3.4 (2024), pp. 566–578.

[44] Rammanohar Das and Raghav Sandhane. "Artificial intelligence in cyber security". In: *Journal of Physics: Conference Series*. Vol. 1964. 4. IOP Publishing. 2021, p. 042072.

[45] Ramya Chinnasamy, Malliga Subramanian, et al. "Deep Learning-driven Methods for Network-based Intrusion Detection Systems: A Systematic Review". In: *ICT Express* (2025).

[46] Karan Napanda, Harsh Shah, et al. "Artificial intelligence techniques for network intrusion detection". In: *International Journal of Engineering Research & Technology (IJERT), ISSN* (2015), pp. 2278–0181.

[47] Debanjana Niogi, Dr Devender Kumar, et al. "Recent Advances and Future Directions in Ai-Based Intrusion Detection Systems for Network Security". In: *Available at SSRN 4485452* (2023).

[48] Chigozie K Ejeofobiri, Olayinka Olubola Victor-Igun, et al. "AI-Driven Secure Intrusion Detection for Internet of Things (IOT) Networks". In: *Asian Journal of Mathematics and Computer Research* 31.4 (2024), pp. 40–55.

[49] Osvaldo Arreche, Tanish Guntur, et al. "Xai-ids: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems". In: *Applied Sciences* 14.10 (2024), p. 4170.

[50] F Tang, M Haider, et al. "MSR107 Estimating Optimal Personalized Treatment Sequencing for Patients with Multiple Myeloma Using Reinforcement Learning". In: *Value in Health* 26.12 (2023), S413.

[51] Kamal Berahmand, Fatemeh Daneshfar, et al. "Autoencoders and their applications in machine learning: a survey". In: *Artificial Intelligence Review* 57.2 (2024), p. 28.

[52] Gulshan Kumar, Krishan Kumar, et al. "The use of artificial intelligence based techniques for intrusion detection: a review". In: *Artificial Intelligence Review* 34 (2010), pp. 369–387.

[53] Jyoti Khurana, Vachali Aggarwal, et al. "A comparative study of deep learning models for network intrusion detection". In: *International Journal of Computer Application* 174.23 (2021), pp. 38–46.

[54] Abiodun Ayantayo, Amrit Kaur, et al. "Network intrusion detection using feature fusion with deep learning". In: *Journal of Big Data* 10.1 (2023), p. 167.

[55] Leila Mohammadpour, Teck Chaw Ling, et al. "A survey of CNN-based network intrusion detection". In: *Applied Sciences* 12.16 (2022), p. 8162.

[56] Saad Albawi, Oguz Bayat, et al. "Social touch gesture recognition using convolutional neural network". In: *Computational Intelligence and Neuroscience* 2018.1 (2018), p. 6973103.

[57] Ali Alferaidi, Kusum Yadav, et al. "Distributed Deep CNN-LSTM Model for Intrusion Detection Method in IoT-Based Vehicles". In: *Mathematical Problems in Engineering* 2022.1 (2022), p. 3424819.

[58] Ibomoiye Domor Mienye, Theo G Swart, et al. "Recurrent neural networks: A comprehensive review of architectures, variants, and applications". In: *Information* 15.9 (2024), p. 517.

[59] Zakir Mujeeb Shaikh and Suguna Ramadass. "Unveiling deep learning powers: LSTM, BiLSTM, GRU, BiGRU, RNN comparison". In: *Indonesian Journal Of Electrical Engineering And Computer Science* 35.1 (2024), pp. 263–273.

[60] Benyamin Ghojogh and Ali Ghodsi. "Recurrent neural networks and long short-term memory networks: Tutorial and survey". In: *arXiv preprint arXiv:2304.11461* (2023).

[61] Yoshua Bengio, Patrice Simard, et al. "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[62] Anvardh Nanduri and Lance Sherry. "Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)". In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. Ieee. 2016, pp. 5C2–1.

[63] Sajan Patel, Brent Griffin, et al. "Predicting future lane changes of other highway vehicles using RNN-based deep models". In: *arXiv preprint arXiv:1801.04340* (2018).

[64] Deepika Singh, Erinc Merdivan, et al. "Human activity recognition using recurrent neural networks". In: *Machine Learning and Knowledge Extraction: First IFIP TC 5, WG 8.4, 8.9, 12.9 International Cross-Domain Conference, CD-MAKE 2017, Reggio, Italy, August 29–September 1, 2017, Proceedings 1*. Springer. 2017, pp. 267–274.

[65] Yuxin He, Ping Huang, et al. "In-depth insights into the application of recurrent neural networks (rnns) in traffic prediction: A comprehensive review". In: *Algorithms* 17.9 (2024), p. 398.

[66] Asmaa A Elsaeidy, Nishant Jagannath, et al. "Replay attack detection in smart cities using deep learning". In: *Ieee Access* 8 (2020), pp. 137825–137837.

[67] Sireesha Kakulla and Srinivas Malladi. "Sybil attack detection in vanet using machine learning approach". In: *Ingenierie des Systemes d'Information* 27.4 (2022), p. 605.

[68] Kyunghyun Cho, Bart Van Merriënboer, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[69] Chunxu Chai, Chuanxiang Ren, et al. "A Multifeature Fusion Short-Term Traffic Flow Prediction Model Based on Deep Learnings". In: *Journal of Advanced Transportation* 2022.1 (2022), p. 1702766.

[70] Ya Zhang, Ravie Chandren Muniyandi, et al. "A Review of Deep Learning Applications in Intrusion Detection Systems: Overcoming Challenges in Spatiotemporal Feature Extraction and Data Imbalance". In: *Applied Sciences* 15.3 (2025), p. 1552.

[71] Stefano Longari, Daniel Humberto Nova Valcarcel, et al. "CANnolo: An anomaly detection system based on LSTM autoencoders for controller area network". In: *IEEE Transactions on Network and Service Management* 18.2 (2020), pp. 1913–1924.

[72] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[73] Ibomoiye Domor Mienye and Nobert Jere. "Deep learning for credit card fraud detection: A review of algorithms, challenges, and solutions". In: *IEEE Access* (2024).

[74] Roberto Cahuantzi, Xinye Chen, et al. "A comparison of LSTM and GRU networks for learning symbolic sequences". In: *Science and Information Conference*. Springer. 2023, pp. 771–785.

[75] Hanting Zhou, Wenhe Chen, et al. "Trustworthy and intelligent fault diagnosis with effective denoising and evidential stacked GRU neural network". In: *Journal of Intelligent Manufacturing* 35.7 (2024), pp. 3523–3542.

[76] Farhad Mortezapour Shiri, Thinagaran Perumal, et al. "A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU". In: *arXiv preprint arXiv:2305.17473* (2023).

[77] Yi-Ying Zhang, Jing Shang, et al. "A self-learning detection method of Sybil attack based on LSTM for electric vehicles". In: *Energies* 13.6 (2020), p. 1382.

[78] Xiaopeng Kan, Zhihong Zhou, et al. "Research on Anomaly Detection in Vehicular CAN Based on Bi-LSTM". In: *Journal of Cyber Security and Mobility* (2023), pp. 629–652.

[79] Ling Xing, Kun Wang, et al. "Intrusion detection method for internet of vehicles based on parallel analysis of spatio-temporal features". In: *Sensors* 23.9 (2023), p. 4399.

[80] Xueli Wang and Qin Wang. "An abnormal traffic detection method using GCN-BiLSTM-Attention in the internet of vehicles environment". In: *EURASIP Journal on Wireless Communications and Networking* 2023.1 (2023), p. 70.

[81] Ankit Manderna, Sushil Kumar, et al. "Vehicular network intrusion detection using a cascaded deep learning approach with multi-variant metaheuristic". In: *Sensors* 23.21 (2023), p. 8772.

[82] Abdul Rehman Javed, Saif Ur Rehman, et al. "CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU". In: *IEEE transactions on network science and engineering* 8.2 (2021), pp. 1456–1466.

[83] Yangyang Song, Nurbol Luktarhan, et al. "TGA: a novel network intrusion detection method based on TCN, BiGRU and attention mechanism". In: *Electronics* 12.13 (2023), p. 2849.

[84] Safi Ullah, Muazzam A Khan, et al. "HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles". In: *Sensors* 22.4 (2022), p. 1340.

[85] Kanika Aggarwal. "Enhancing Intrusion Detection in Vehicular Networks through Deep Learning Approaches". M.A.Sc. Thesis. École de technologie supérieure, Université du Québec, 2023.

[86] Pengzhou Cheng, Kai Xu, et al. "TCAN-IDS: intrusion detection system for internet of vehicle using temporal convolutional attention network". In: *Symmetry* 14.2 (2022), p. 310.

[87] Microsoft. *Visual Studio Code*. `https://code.visualstudio.com`. Accessed: 2024-01-17.

[88] Python Software Foundation. *Python Programming Language*. `https://www.python.org`. Accessed: 2024-01-17.

[89] Rens W Van Der Heijden, Thomas Lukaseder, et al. "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets". In: *Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part I*. Springer. 2018, pp. 318–337.

[90] Rens van der Heijden, Stefan Dietzel, et al. *VeReMi Dataset: Vehicular Reference Misbehavior Dataset*. `https://veremi-dataset.github.io`. Accessed: 2024-02-17.

[91] Shilpasayura. *Veremi Extension Simple Dataset*. `https://www.kaggle.com/datasets/shilpasayura/veremi-extension-simple`. Accessed: 2024-04-20.

Abstract

As vehicular networks evolve, so do the threats targeting them, particularly those that exploit identity mechanisms in subtle, persistent ways. This study addresses identity-based attacks in the Internet of Vehicles (IoV), focusing on Advanced Persistent Threat (APT)-like behaviors such as Sybil and replay attacks, which manipulate identity systems to evade detection. Using the Vehicular Misbehavior Detection (VeReMi) dataset, which simulates vehicular attack scenarios, we preprocess and reinterpret its attacks to align with identity misuse patterns, despite the dataset's lack of explicit long-term APT stages or real identity data. We propose a hybrid deep learning model that combines Temporal Convolutional Network (TCN)s, Transformers, Bidirectional Gated Recurrent Unit (BiGRU)s, and Squeeze-and-Excitation (SE) blocks to capture both short-term and long-term behavioral anomalies. The model is evaluated against benchmark architectures, with emphasis on minimizing false negatives, a critical requirement for IoV security. Experimental results demonstrate strong performance, achieving a 98.0% accuracy and a 0.9% false negative rate, highlighting its effectiveness in detecting identity-based threats. This work contributes practical insights for deploying adaptive, identity-aware Intrusion Detection System (IDS) in vehicular networks, bridging the gap between theoretical research and real-world applications.

Keywords: Intrusion Detection Systems (IDS), Advanced Persistent Threats (APT), Identity-based Attacks, Anomaly Detection, Internet of Vehicles (IoV), Vehicular Ad-hoc Networks (VANETs), V2X Security, Deep Learning, Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), BiGRU (Bidirectional Gated Recurrent Unit), Transformer Encoder, Temporal Convolutional Network (TCN), Squeeze-and-Excitation (SE) Block, Hybrid Deep Learning Models, VeReMi Dataset, VeReMi Extension (Simple), Supervised Learning, Identity-based APT Detection, AI-driven Security Solutions.

Résumé

Les attaques basées sur l'identité dans l'Internet des véhicules représentent une nouvelle classe de menaces avancées, persistantes et furtives, exploitant les systèmes d'identité pour contourner les mécanismes de sécurité. Ces attaques ne suivent pas toujours des modèles d'intrusion classiques mais adoptent des comportements discrets, comme la rotation de pseudonymes, la falsification de données ou la relecture de messages, rendant leur détection particulièrement difficile. Dans ce contexte, nous proposons un système intelligent de détection d'intrusions capable d'identifier ces comportements de type APT (menace persistante avancée), en se basant uniquement sur des caractéristiques comportementales temporelles, sans utiliser d'identifiants explicites. Notre approche repose sur un modèle hybride combinant un réseau de neurones convolutionnels temporels, des transformers, un GRU bidirectionnel et des mécanismes d'attention. Le modèle est entraîné sur une version étendue du dataset VeReMi, qui, bien qu'il ne contienne pas directement des attaques APT longues ou des identifiants réels, inclut des types d'attaques tels que Sybil, relecture de données, falsification de position, qui simulent les comportements typiques des attaques par usurpation d'identité à long terme dans les réseaux véhiculaires. Les résultats expérimentaux montrent que notre modèle atteint une précision de 98%, avec un taux de faux négatifs de 0,9%, démontrant son efficacité dans la détection de comportements anormaux liés à l'identité, tout en maintenant un faible taux d'erreur. Cette recherche fournit ainsi une base concrète pour des systèmes de détection d'intrusion intelligents orientés vers l'analyse comportementale dans les réseaux véhiculaires, en vue d'une application réelle.

Mots Clés : Systèmes de détection d'intrusion, Menaces persistantes avancées, Attaques basées sur l'identité, Détection d'anomalies, Internet des véhicules, Réseaux véhiculaires ad hoc (VANETs), Sécurité V2X (véhicule-à-tout), Apprentissage profond, Réseaux de neurones récurrents, Réseaux de neurones convolutifs, Unité récurrente à portes bidirectionnelle, Encodeur Transformer, Réseau convolutif temporel, Bloc Squeeze-and-Excitation, Modèles hybrides d'apprentissage profond, Jeu de données VeReMi, Extension simple de VeReMi, Apprentissage supervisé, Détection d'APT basées sur l'identité, Solutions de sécurité basées sur l'IA.