**A**

**Major Project Report**

**On**

**INNOVATIVE INTRUSION DETECTION SYSTEMS FOR ENHANCING SECURITY IN INTERNET OF VEHICLES USING ADVANCED MACHINE LEARNING TECHNIQUES**

**Submitted in the partial fulfilment of the award of the**

**Degree of Bachelor of Technology**

**In**

**COMPUTER SCIENCE ENGINEERING**

SUBMITTED BY

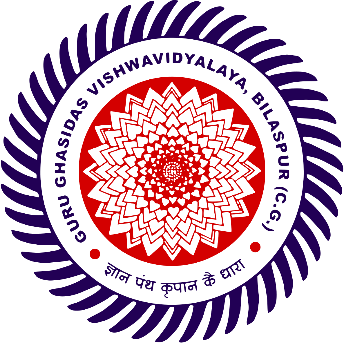
**Abhay Sonu Thakur (21027101)**

**Ajay Kumar Sharma (21027106)**

UNDER THE GUIDANCE OF

Mr. Satish Kumar Negi

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SCHOOL OF STUDIES OF ENGINEERING & TECHNOLOGY**

**GURU GHASIDAS VISHWAVIDYALAYA, BILASPUR, (C.G), INDIA**

**April 2025**

# 

# **CERTIFICATE**

We hereby certify that the work which is being presented in the B.Tech. Major Project Report entitled “**Innovative Intrusion Detection Systems for Enhancing Security in Internet of Vehicles Using Advanced Machine Learning Techniques**” in partial fulfillment of the requirements for the major project of the Bachelor of Technology in Computer Science and Engineering and submitted to the Department of Computer Science and Engineering, School of Studies of Engineering And Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh, India is an authentic record of my own work carried out during the period of December 2024 - April 2025 (8th semester) under the supervision of Mr. Satish Kumar Negi, Assistant Professor of CSE Department.

The matter presented in this Project Report has not been submitted by me or by anyone else for the award of any other degree elsewhere.

**NAME OF STUDENTS SIGNATURE**

**AJAY KUMAR SHARMA (21027106) ………………**

**ABHAY SONU THAKUR (21027101) ...…………….**

This is to certify that the above statement made by the student(s) is correct to the best of our knowledge.

Signature of Supervisor

**Mr. Satish Kumar Negi**

Assistant Professor

**Prof. Manjunathswamy B.E**

Head of the Department,

Computer Science and Engineerng Department

# **DECLARATION**

We at this moment declare that the project entitled – “**Innovative Intrusion Detection Systems for Enhancing Security in Internet of Vehicles Using Advanced Machine Learning Techniques**”, which is being submitted as a Major Project of the 8th Semester to Department of Computer Science & Engineering Guru Ghasidas Vishwavidyalaya, Bilaspur (C.G.) is an authentic record of our genuine work done under the guidance of **Mr. Satish Kumar Negi,** Assistant Professor,Dept. of Computer Science & Engineering, School of Studies of Engineering & Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh. We also declare that if any information we provide is false, we shall be held responsible for the consequences.

**Name of Students Signature**

**1. Ajay Kumar Sharma (**21027106**) …………………**

**2. Abhay Sonu Thakur (**21027101**) …………………**

# **ACKNOWLEDGEMENT**

We express our sincere gratitude to **Mr. Satish Kumar Negi**, Assistant Professor, Department of Computer Science and Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India, for his stimulating guidance, continuous encouragement, and supervision throughout the course of the present work.

We would like to place on record our deep sense of gratitude to **Prof. Manjunathswamy B. E.**, Professor and Head, Department of Computer Science and Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India, for his generous guidance, help, and useful suggestions.

We are extremely thankful to **Prof. Sharad Chandra Srivastava**, Dean, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India, for providing us with the infrastructural facilities to work in, without which this work would not have been possible.

We also wish to extend our sincere thanks to **Prof. Alok Kumar Singh Kushwaha, Mr. Pushpendra Kumar Chandra,** Assistant Professor, and **Dr. Amit Kumar Baghel,** Assistant Professor, for their valuable support, guidance, and encouragement throughout the project.

Lastly, we express our gratitude to the entire faculty and our classmates for their constant inspiration, encouragement, insightful comments, and constructive suggestions which greatly improved the quality of this project work.

**Name of Students Signature**

**1. Ajay Kumar Sharma (**21027106**) ……………………………………**

**2. Abhay Sonu Thakur (**21027101**) .……………………………………**

**ABSTRACT**

The Internet of Vehicles (IoV) is reshaping how vehicles interact with their surroundings by enabling real-time communication between vehicles, roadside infrastructure, and cloud-based systems. While this connectivity improves road safety, traffic efficiency, and overall driving experience, it also introduces serious cybersecurity risks. Threat actors can exploit vulnerabilities in the network to launch various attacks, putting both data integrity and physical safety at risk. In such a dynamic and open environment, Intrusion Detection Systems (IDS) play a critical role in identifying malicious activity before damage is done. This project explores the design and performance of two different IDS models Random Forest (RF) and Long Short-Term Memory (LSTM) to detect anomalies in IoV networks. To build a robust dataset that represents diverse network conditions and attack types, we merged the CICIDS2017 and CICIDS2018 datasets using fuzzy feature alignment. From this merged dataset, we created two classification tasks: binary (benign vs. attack) and multiclass (specific types of attacks). Each model was trained and evaluated independently on both classification problems. The Random Forest model, known for its simplicity and high-speed execution, showed excellent performance in detecting intrusions with low computational overhead. On the other hand, the LSTM model demonstrated a higher capacity to understand sequential patterns in traffic data, which is especially useful for identifying sophisticated, time-dependent threats like Botnet or slow DDoS attacks. Our findings show that both models are effective in their own right: RF is suitable for real-time applications with limited processing power, while LSTM provides higher accuracy where time-series behaviour matters. This comparative study helps lay a foundation for implementing adaptive, efficient IDS solutions tailored to the unique challenges of IoV environments.

**TABLE OF CONTENTS**

[CERTIFICATE i](#_Toc197341210)

[DECLARATION ii](#_Toc197341212)

[ACKNOWLEDGEMENT iii](#_Toc197341213)

[ABSTRACT iv](#_Toc197341214)

[LIST OF TABLES vii](#_Toc197341215)

[LIST OF FIGURES viii](#_Toc197341216)

[ABBREVIATION ix](#_Toc197341217)

[**CHAPTER 1 INTRODUCTON** 1](#_Toc197341218)

[1.1 Introduction to IOV 2](#_Toc197341219)

[1.2 Definition and Background 2](#_Toc197341220)

[1.3 Security Issues 4](#_Toc197341221)

[1.4 Intrusion Detection Systems (IDS) 7](#_Toc197341222)

[**CHAPTER 2 LITERATURE REVIEW** 16](#_Toc197341223)

[2. Literature Overview 17](#_Toc197341224)

[**CHAPTER 3 DATASET** 24](#_Toc197341225)

[3. Dataset Overview 25](#_Toc197341226)

[3.1 CICIDS2017 Dataset 25](#_Toc197341227)

[3.2 CICIDS2018 Dataset 27](#_Toc197341228)

[3.3 Data Processing 30](#_Toc197341229)

[**CHAPTER 4 METHODOLOGY** 32](#_Toc197341230)

[4. Methodology of Project 33](#_Toc197341231)

[4.1. Data Acquisition and Preprocessing 33](#_Toc197341232)

[4.2. Model Selection and Design 34](#_Toc197341233)

[4.2.1 Random Forest (RF) 34](#_Toc197341234)

[4.2.2 Long Short-Term Memory (LSTM) 35](#_Toc197341235)

[4.3. Training the Models And Model Evaluation 38](#_Toc197341236)

[4.3.1 Training the Random Forest Model for binary Classification 39](#_Toc197341237)

[4.3.2 Training the Random Forest Model for Multiclass Classification 40](#_Toc197341238)

[4.3.3 Training the LSTM Model for Multiclass Classification 41](#_Toc197341240)

[4.3.4 Training the LSTM Model for Binary Classification 43](#_Toc197341242)

[**CHAPTER 5 RSULT AND DISCUSSION** 46](#_Toc197341244)

[5. Result And Discussion 47](#_Toc197341245)

[5.1 Random Forest: Binary Classification 47](#_Toc197341246)

[5.2 Random Forest: Multiclass Classification 48](#_Toc197341247)

[5.3 LSTM: Binary Classification 50](#_Toc197341248)

[5.4 LSTM: Multiclass Classification 51](#_Toc197341249)

[**CHAPTER 6 CONCLUSION & FUTURE WORK** 54](#_Toc197341251)

[6.1 Conclusion 55](#_Toc197341252)

[6.2 Future Work 55](#_Toc197341253)

[**REFERENCES** 57](#_Toc197341254)

# **LIST OF TABLES**

Table 1.1 Comparison between different Network Protocols for intra-Network……………....5

Table 1.2 Comparison between different Network Protocols for inter-Network……………....5

Table 1.3 Dataset used for IoV IDS in Literature……………………………………………...9

Table 1.4 Security Attacks in IoV Network………………………………………………….10

Table 2.1 Studies on Intrusion Detection Systems for IoV…………………………………..17

Table 3.1 All Features in CICIDS2017……………………………………………………....26

Table 3.2 All Features in CICIDS2018………………………………………………………28

Table 4.1 Difference between RF and LSTM………………………………………………..38

# **LIST OF FIGURES**

Figure 1.1 Security vulnerabilities in Internet of Vehicle (IoV)……………………………….6

Figure 1.2 IDS Approaches……………………………………………………………………8

Figure1.3 IoV Security challenges…………………………………………………………...14

Figure 2.1 Flowchart of Intrusion Detection Process…………………………………..…….20

Figure 2.2 Flowchart of Different Detection In IDS……………………………………...….21

Figure 2.3 Framework MTH………………………………………………………………....22

Figure3.1 Dat Processing Funnel…………………………………………………………….30

Figure4.1 Network Intrusion Detection Data Processing……………………………………34

Figure5.1 Output of Random Forest: Binary Classification…………………………………47

Figure5.2 Confusion Matrix of Random Forest: Binary Classification………………….….48

Figure5.3 Output of Random Forest: Multiclass Classification …………………………….49

Figure5.4 Confusion Matrix of Random Forest: Multiclass Classification………………....50

Figure5.5 Output of LSTM: Binary Classification………………………………………….50

Figure5.6 Loss in LSTM Multiclass classification………………………………………….51

Figure5.7 Confusion Matrix of LSTM: Multiclass Classification…………...……………...51

Figure5.8 Output of LSTM: Multi Classification…………………………………………...52

Figure 5.9 Comparison of Accuracy in LSTM and RF……………………………………..52

Figure 5.10 Comparison of F1-Score in LSTM and RF…………………………………….53

Figure 5.11 Comparison of Accuracy of different mode in LSTM and RF………………...53

Figure 5.12 Comparison of F1-Score of different mode in LSTM and RF…………………53

# **ABBREVIATION**

|  |  |
| --- | --- |
| IDS | Intrusion Detection System |
| IoV | Internet of Vehicles |
| ECU | Electronic Control Unit |
| 5G | Fifth Generation (Mobile Network) |
| V2X | Vehicle-to-Everything |
| VANET | Vehicular Ad Hoc Network |
| IoT | Internet of Things |
| V2V | Vehicle-to-Vehicle |
| V2I | Vehicle-to-Infrastructure |
| V2P | Vehicle-to-Pedestrian |
| V2G | Vehicle-to-Grid |
| ITS | Intelligent Transportation System |
| CAN | Controller Area Network |
| DoS | Denial of Service |
| LTE-V | Long-Term Evolution Vehicular Network |
| DSRC | Dedicated Short-Range Communication |
| DDoS | Distributed Denial of Service |
| LSTM | Long Short-Term Memory |
| AI | Artificial Intelligence |
| ML | Machine Learning |
| CNN | Convolutional Neural Network |
| TCN | Temporal Convolutional Network |
| TCAN-IDS | Temporal CAN Intrusion Detection System |

# 

# **CHAPTER 1**

**INTRODUCTION**

## **1.1 Introduction to IOV**

The Internet of Vehicles (IoV) is reshaping the future of transportation by enabling vehicles to communicate not only with each other but also with surrounding infrastructure. Equipped with advanced sensors and Electronic Control Units (ECUs), these smart vehicles are paving the way toward fully autonomous driving [1]. Rapid advancements in wireless communication technologies, such as 5G and V2X (Vehicle-to-Everything) communications, have further enhanced data transfer speeds and reliability, reducing latency and improving overall system performance [2]. By merging Vehicular Ad Hoc Networks (VANETs) with the Internet of Things (IoT), the IoV ecosystem delivers essential services such as real-time traffic updates, navigation assistance, and smart safety features [3].

To achieve seamless connectivity, IoV utilizes a range of networking technologies that link internal vehicle systems with external infrastructure. According to Gartner's research, 5G IoT will play a critical role in this transformation, with projections indicating that connected vehicles will account for nearly 53% of all 5G IoT endpoints by 2030 [4] highlighting a major growth opportunity within the automotive sector.

However, this extensive connectivity also introduces serious security challenges. Continuous interactions between vehicles and road infrastructure make IoV networks vulnerable to cyber intrusions [5]. Ensuring robust security is vital, as any compromise could threaten the safety of drivers, passengers, and pedestrians alike. Cyberattacks could lead to vehicle hijacking, the spread of false information, or breaches of sensitive data [6]. Therefore, proactive measures to detect, prevent, and address these risks are absolutely essential to maintain trust and ensure safety within the IoV environment.

## **1.2 Definition and Background**

The **Internet of Vehicles (IoV)** is a complex network that supports data from connected cars and **Vehicular Ad Hoc Networks (VANETs)** [1]. This system enables essential communication between sensors and **Electronic Control Units (ECUs)** within vehicles, facilitating autonomous driving, real-time information sharing, and in-car entertainment [2]. IoV also promotes smoother driving and more efficient traffic management by supporting data exchange among various road entities [3].

The IoV communication system is divided into two main categories:

**1.2.1 Intra-Vehicle Communication**: This involves all exchanges between sensors and ECUs within the vehicle. Often called "vehicle-to-sensor" communication, it coordinates vehicle states and actions [4]. Each ECU has its own sensor and actuator, enabling specific functions in three main areas:

* **Body domain**: Controls lights, windows, and mirrors.
* **Chassis domain**: Manages real-time safety systems like brakes and suspension.
* **Powertrain domain**: Controls the engine, transmission, and safety features like traction control [5].

For driving safety, frequent data exchange across these domains is crucial. Intra-vehicle communication is now standard in high-end vehicles, where approximately 70 ECUs may relay up to 2,500 signals internally [6].

**1.2.2 Inter-Vehicle Communication**: This type covers exchanges between vehicles and other entities on the road. Inter-vehicle communication includes:

* **Vehicle-to-Vehicle (V2V)**: Allows drivers to share traffic updates and alert others to nearby hazards [7].
* **Vehicle-to-Infrastructure (V2I)**: Supplies real-time traffic, weather, and security alerts to drivers and can provide network access [8].
* **Vehicle-to-Pedestrian (V2P)**: Connects vehicles and pedestrians through smartphones. For instance, the **U.S. Department of Transportation** has proposed apps to help visually impaired pedestrians interact with traffic signals [9].
* **Vehicle-to-Grid (V2G)**: Supports connections between electric vehicles and the grid for efficient charging and battery management [10].

Collectively known as **Vehicle-to-Everything (V2X)**, these networks integrate vehicles, pedestrians, infrastructure, and smart devices into an **Intelligent Transportation System (ITS)**, aiming to improve safety, traffic flow, and the overall driving experience [11].

## **1.3 Security Issues**

Network connectivity always brings some level of risk, and the **Internet of Vehicles (IoV)** is no exception. With countless sensors and processors managing and transmitting data, IoV becomes vulnerable to attacks [1]. The multiple communication channels, like **Vehicle-to-Vehicle (V2V)** and **Vehicle-to-Infrastructure (V2I)**, can make IoV systems an appealing target for intruders [2].

Security issues in IoV are critical, as any false or malicious information could disrupt a vehicle’s decision-making processes, potentially leading to harmful consequences [3]. Additionally, vehicles gather sensitive user data, such as location and credentials, so any privacy breach could erode user trust and damage a manufacturer’s reputation [4]. IoV security threats focus on compromising the **confidentiality, integrity, availability, and authenticity** of the network, generally falling into two categories: **inter-vehicle attacks** and **intra-vehicle attacks** [5].

**1.3.1 Intra-Vehicle Network**

In intra-vehicle networks, various protocols are used to manage communication, with the **CAN (Controller Area Network) bus** being the most common in cars today [6]. Its popularity comes from being affordable, easy to install, and highly efficient for real-time data transfer. However, the CAN bus has some significant security issues. Without built-in authentication or encryption, it’s vulnerable to attacks like **Denial of Service (DoS)** and **data injection** [7]. Since CAN broadcasts data without encryption, any node-whether legitimate or malicious-can send information across the network, creating an easy entry point for hackers. **Manipulation attacks** are especially common, where attackers craft specific messages to disrupt the vehicle’s normal functions and compromise its safety [8].

**1.3.2 Inter-Vehicle Network**

**V2X (Vehicle-to-Everything)** technology enables communication between vehicles and various road entities, such as infrastructure, pedestrians, and even the power grid, with the goal of improving driving experiences and safety [9]. However, this interconnected network presents a vulnerability. Malicious actors can exploit these communication links to access sensitive information, disrupt the system, or block secure data transmission [10].

Vehicles connect to external devices, like smartphones, through communication protocols such as **LTE-V (Long-Term Evolution Vehicular Network)**, **DSRC (Dedicated Short-Range Communication)**, and other global mobile communication systems [11]. This reliance on these channels exposes vehicles to network attacks, including **Distributed Denial of Service (DDoS)**, **data modification**, **routing attacks**, **phishing**, and **Man-In-The-Middle (MITM)** attacks, among others [12].

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Protocol** | **Bit-rate** | **Application** | **Domain** | **Standard** |
| CAN | 40 Kbps - 250 Kbps | Real-time critical applications Non-critical application | Powertrain, Body, Chassis | ISO 11898 |
| LIN | 1-20 Kb/s | Non-critical applications | Body | ISO 17987 |
| Flex Ray | 10-20 Mb/s | Critical applications | Powertrain and Chassis | ISO 17458 |

**Table1.1 Comparison between different Network Protocols for intra-Network**

This Table 1.1 Compares different communication protocols used in vehicles: CAN, LIN, and Flex Ray. These are responsible for allowing various electronic systems inside a car to talk to each other efficiently.

CAN (Controller Area Network) is used for both critical and non-critical tasks. Its data transfer speed ranges from 40 Kbps to 250 Kbps, making it suitable for real-time systems like engine control as well as less urgent functions like controlling the windows. It’s commonly used in the powertrain, body, and chassis sections of a vehicle, and it follows the ISO 11898 standard.

LIN (Local Interconnect Network) operates at slower speeds, between 1 to 20 Kbps, and is mainly used for non-critical systems such as seat adjustment or interior lighting. It’s typically applied in the body area of the car and follows the ISO 17987 standard.

Flex Ray offers high-speed communication between 10 to 20 Mbps and is meant for critical applications that require quick and reliable data exchange, like advanced braking or steering systems. It is used in the powertrain and chassis and is standardized under ISO 17458.

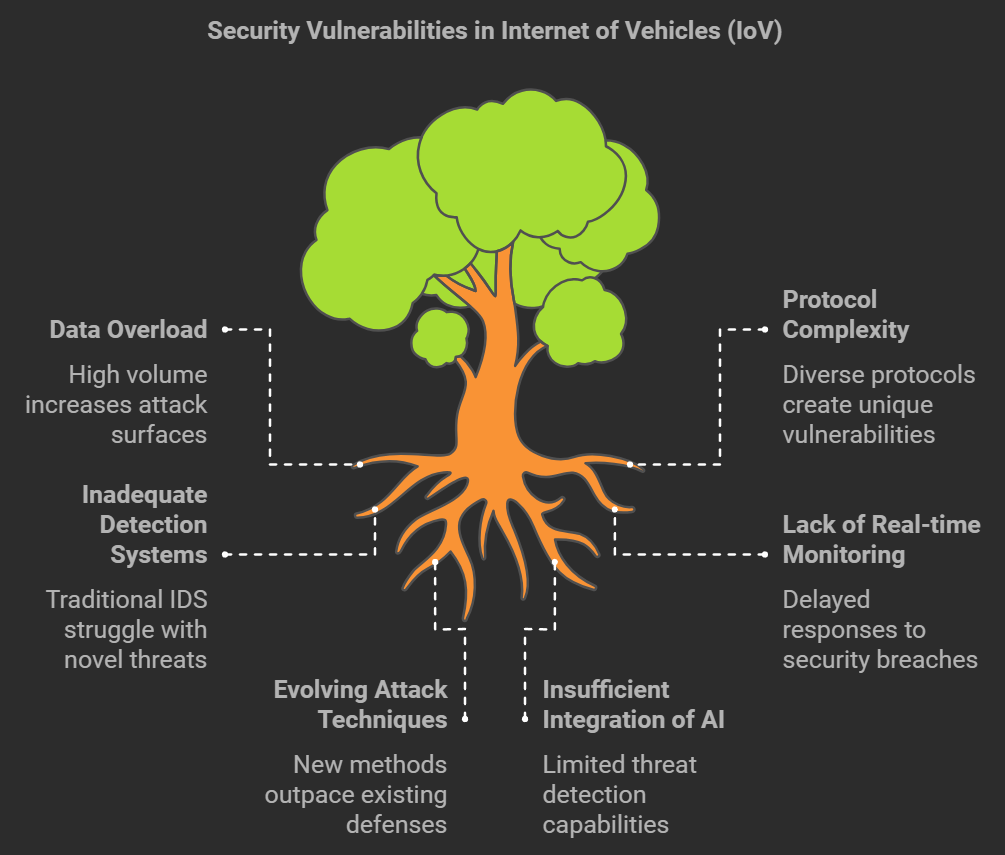
**Table1.2 Comparison between different Network Protocols for inter-Network**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Protocol** | **Bit Rate** | **Range** | **Delay** | **Application** | **Standard** |
| DSRC | 1-27 Mbps | 30 m -1000 m | 150 ms | Safety and Non-Safety | IEEE 802.11p |
| LTE-V | 1 Gbps | < 2000 m | 50 ms | Non-Safety | LTE-V |
| 5G | 10 Gbps | < 2000 m | 1 ms | Safety and Non-Safety | N/A |

This **table 1.2** compares three different wireless communication technologies used in vehicle-to-everything (V2X) systems: DSRC, LTE-V, and 5G. These technologies are essential for enabling vehicles to communicate with each other and with surrounding infrastructure to improve safety, traffic flow, and overall driving experience.

DSRC (Dedicated Short-Range Communication) offers data speeds between 1 to 27 Mbps and covers a range of 30 meters to 1000 meters. It has a delay of about 150 milliseconds, making it suitable for both safety and non-safety applications. DSRC follows the IEEE 802.11p standard and has been widely used in earlier V2X systems.

LTE-V (Long-Term Evolution for Vehicles) is capable of much higher speeds, up to 1 Gbps, and supports communication within a 2000-meter range. Its latency is lower than DSRC at around 50 milliseconds, making it more efficient for non-safety applications, like infotainment and vehicle tracking. It uses the LTE-V standard.

5G brings the fastest speeds of all, up to 10 Gbps, and maintains the same <2000 m range as LTE-V. Its key advantage is its ultra-low delay of only 1 millisecond, which makes it highly suitable for both safety-critical and non-safety tasks, such as autonomous driving and high-speed vehicle coordination. As it is still evolving, it does not have a single associated standard listed in the table.

**Figure 1.1 Security vulnerabilities in Internet of Vehiclwa(IoV)**

The diagram highlights key security challenges in the Internet of Vehicles (IoV). These include data overload, complex communication protocols, lack of real-time monitoring, and outdated detection systems. At the root of these issues are evolving cyberattack methods and limited use of AI, making it difficult to keep vehicle networks secure.

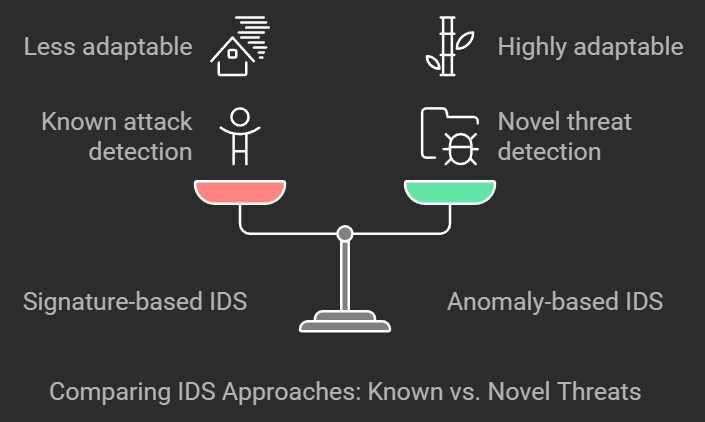
## **1.4 Intrusion Detection Systems (IDS)**

The Internet of Vehicles (IoV) is an emerging paradigm that is transforming the way vehicles communicate and interact with each other, infrastructure, and other entities on the road [1]. This interconnected ecosystem opens up numerous opportunities for improving road safety, enhancing driving experiences, and optimizing traffic management [2]. By leveraging sensory data, vehicles can exchange real-time information on road conditions, vehicle status, and other vital metrics, facilitating a more synchronized and intelligent transportation system [3]. However, with the rise of these advanced technologies, the IoV also introduces a range of security vulnerabilities. The vast amount of data exchanged between vehicles and their networks can potentially be exploited by malicious actors, jeopardizing the efficiency of communication channels and compromising the safety of the system [4].

To mitigate these security risks, effective Network Intrusion Detection Systems (IDS) are essential. IDS play a crucial role in monitoring network traffic for any signs of malicious activity, identifying potential security threats, and helping to maintain the integrity of communications across IoV networks [5]. Recent research in the IoV space has focused on enhancing these systems, particularly through **anomaly-based detection methods** that utilize traditional **Machine Learning (ML)** algorithms to identify irregular patterns in network traffic [6]. This approach is increasingly important in an environment where traditional security methods may struggle to cope with the complexity and scale of the IoV [7]. In the IoV environment, various network protocols are employed for communication between vehicles, each with distinct characteristics and susceptibilities. The choice of protocol, as well as the nature of the attacks it may face, significantly influences the design of intrusion detection techniques [8]. For instance, in the context of **vehicle-to-vehicle (V2V)** and **vehicle-to-infrastructure (V2I)** communication, protocols like **Long-Term Evolution for Vehicles (LTE-V)** and the **Controller Area Network (CAN)** bus are commonly used [9]. LTE-V, which is designed for high-speed, mobile communication, typically involves more complex packet structures and higher communication overhead [10]. On the other hand, the CAN bus protocol, which is widely used for communication within vehicles, is simpler and more constrained in terms of data fields [11]. This simplicity, however, makes it more vulnerable to targeted attacks on specific fields, such as the Data and Arbitration IDs [12].

Given these differences, research has led to the development of IDS solutions tailored specifically for inter-vehicle and intra-vehicle networks, accounting for the unique complexities of each communication type [13]. Some studies have focused on creating intrusion detection systems that are specialized for the CAN bus protocol, while others take a broader, cross-network approach, proposing frameworks capable of detecting intrusions across multiple network types within the IoV [14].

IDS can be broadly categorized into two types: **signature-based** and **anomaly-based**. **Signature-based IDS** rely on predefined patterns or "signatures" that represent known attack behaviors [15]. When the system detects network traffic that matches a signature, it flags the activity as a potential threat. While this approach is highly effective in identifying known attacks, it is less capable of detecting novel or modified attack strategies, as it depends on the system having prior knowledge of the threat [16]. Therefore, signature-based IDS are less adaptable to new attack vectors that may emerge in dynamic environments like the IoV.

On the other hand, **anomaly-based IDS** provide a more flexible and adaptive solution. These systems analyze network traffic and establish a baseline of normal behavior [17]. By continuously monitoring the traffic and detecting deviations from this baseline, anomaly-based systems can identify unusual patterns that may indicate the presence of a new or unknown attack, even if there are no existing signatures for it. This ability to detect novel threats makes anomaly-based IDS particularly valuable in dynamic and evolving environments such as the IoV, where new attack techniques can emerge rapidly [18].

**Figure 1.2 IDS Approaches**

With the increasing complexity of IoV networks and the growing volume of data being exchanged, Machine Learning (ML) techniques have become an essential tool for enhancing IDS performance. By incorporating ML algorithms, IDS can improve their ability to detect subtle anomalies and classify suspicious activities based on a wider range of network features. Through supervised or unsupervised learning, these systems can identify patterns that human analysts may overlook, enabling more accurate detection of potential threats.

As the IoV continues to evolve, there is also a growing interest in integrating advanced technologies like blockchain and cryptographic methods to strengthen data privacy and security. Blockchain offers a decentralized, tamper-proof ledger that can help prevent unauthorized access and ensure the integrity of the data exchanged within the IoV. Meanwhile, cryptographic techniques provide an additional layer of security by encrypting sensitive information, making it more difficult for attackers to intercept or alter communications.

Despite these advancements, the need for effective IDS remains critical. In a highly connected IoV environment, it is vital that security systems can quickly detect abnormal behavior and alert authorities or users in real time. The integration of Artificial Intelligence (AI) into IoV networks presents a promising avenue for enhancing the efficiency and scalability of data analysis, enabling better threat detection and a more proactive security posture. As AI-driven Machine Learning and Deep Learning algorithms become increasingly sophisticated, they offer the potential to revolutionize the protection of IoV systems, ensuring that vehicles remain secure and that the future of intelligent transportation is both safe and efficient.

**Table1.3 Dataset used for IoV IDS in Literature**

|  |  |  |
| --- | --- | --- |
| **Network System** | **Dataset** | **Attack Types** |
| Inter-Vehicle | CICIDS2017 | DoS, DDoS, Brute Force, SQL Injection, XSS, Infiltration, Botnet, Port Scanning, Normal Traffic |
| CSE-CIC-IDS2018 | DoS, DDoS, Brute Force, SQL Injection, XSS, Infiltration, Botnet, Port Scanning, Normal Traffic |
| ISCXIDS2012 | DoS, DDoS, Brute Force, Infiltration, Normal Traffic |
| NSL-KDD | DoS, Probing, Remote to Local, User to Root |
| UNSW-NB15 | Fuzzers, Exploits, Worms, Shellcode, Scanning, Generic, DoS, Backdoors, Reconnaissance |
| ToN-IoT | Ransomware, DoS, DDoS, Scanning, Backdoors, Injections, MITM, Password Attacks, Cross-site Scripting |
| KDDCup99 | DoS, Probing, Remote to Local, User to Root |
| IoT BotNet | DDoS, DoS, Normal Traffic, Scanning, Data Theft |
| KDD99 | DoS, Probing, Remote to Local, User to Root |
| AWID | SQL Injection, Malware, Rogue Access Points, SSH, Evil Twin, Botnet, Spoofing, Krack, Deauthentication |
| Intra-Vehicle | Car Hacking | Fuzzy Attacks, DoS, Gear/RPM Spoofing |
| CAN Intrusion | Fuzzy Attacks, DoS, Gear/RPM Spoofing |
| ROAD | Fuzzing, Masquerading, ID Fabrication |

The table 1.3 outlines various publicly available datasets used for analyzing cyberattacks in vehicular networks, categorized into Inter-Vehicle and Intra-Vehicle systems based on the type of communication they represent.

For Inter-Vehicle systems, datasets like CICIDS2017, CSE-CIC-IDS2018, and ISCXIDS2012 focus on a wide range of attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), brute force attacks, SQL injections, and botnets, along with normal traffic data for comparison. Datasets like NSL-KDD and KDDCup99 include older but still relevant attacks, such as probing, remote-to-local, and user-to-root intrusions. UNSW-NB15 expands this with more modern threats including exploits, worms, fuzzing, and reconnaissance. ToN-IoT offers a diverse set of attacks, including ransomware, MITM, and password breaches, making it suitable for smart vehicle IoT environments. The AWID dataset focuses on wireless network attacks relevant to vehicular Wi-Fi environments, such as rogue access points and deauthentication.

In the case of Intra-Vehicle systems, the datasets reflect attacks on internal communication networks like the CAN Bus. Datasets such as Car Hacking, CAN Intrusion, and ROAD capture malicious actions like fuzzing, spoofing of gear or RPM data, and ID fabrication. These attacks are specifically designed to mislead the internal vehicle control units, often leading to potentially dangerous outcomes if not properly detected.

This categorization helps researchers and developers choose the most suitable datasets depending on whether they are focusing on external vehicular communication threats or internal system vulnerabilities.

**Table1.4 Security Attacks in IoV Network**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attack Category** | **Attack Type** | **Execution Method** | **Example** | **Security Goal** |
| Inter-Vehicle | DoS/DDoS | Saturating network with high traffic to disrupt normal operations | Prevent vehicles from accessing critical services like traffic updates or toll info | Availability |
| Modification | Changing data during transmission to/from infrastructure | Sending incorrect data about temperature or battery to mislead system controls | Integrity |
| Selfish | Refusing to share data with other nodes | Refusal to forward data causing disruption in network cooperation | Availability |
| Impersonation | Using fake/multiple identities to send misleading info | Faking identity in EV charging queue to manipulate access | Authentication |
| Eavesdropping | Intercepting communication to gather sensitive data | Masquerading as a legitimate node to capture vehicle location data | Confidentiality |
| Routing | Altering routing information; includes black/gray/wormhole attacks | Disrupt cooperative driving by sending false routing paths | Availability |
| Phishing | Gaining access through fake identities or deceptive login points | Fake Access Point steals sensitive info like credentials | Authentication |
| Rogue Updates | Deploying malicious software masquerading as official updates | Installing untrusted firmware that modifies vehicle behavior | Availability |
| MITM (Man-in-the-Middle) | Gaining control between sender and receiver to manipulate messages | Interfering with GPS data during transmission | Confidentiality |
| Replay | Capturing and replaying valid packets to deceive system | Faking vehicle presence at a location during a past event | Authentication |
| Intra-Vehicle (CAN Bus) | DoS | Flooding CAN network to cause loss of functionality | Blocking communication between ECUs by sending overflow messages | Availability |
| Spoofing | Sending unauthorized CAN messages as legitimate | Injecting false signals to disable or hijack functions like brakes | Integrity, Authentication |
| Fuzzing | Transmitting random CAN frames and observing responses | Inducing system errors or faults through unanticipated inputs | Integrity |
| Falsifying | Tampering with message contents to introduce errors | Altering speed or fuel data shown to the driver | Integrity |

The table 1.4 presents a detailed classification of cyberattacks in the Internet of Vehicles (IoV), divided into inter-vehicle and intra-vehicle (CAN Bus) categories. Inter-vehicle attacks involve communication between vehicles or with roadside infrastructure. For example, Denial of Service (DoS) and Distributed DoS (DDoS) attacks aim to overload the network, making services unavailable. Modification attacks involve tampering with transmitted data, such as altering battery or temperature information. Selfish attacks occur when a node refuses to forward data, reducing overall network efficiency. Impersonation, or Sybil attacks, involve fake identities sending misleading messages. Eavesdropping captures sensitive data by pretending to be a trusted node. Routing attacks mislead the vehicle by altering data paths. Phishing tricks users through fake access points to steal credentials. Rogue updates deliver harmful software disguised as legitimate. Man-in-the-middle (MITM) attacks manipulate data during transmission. Replay attacks reuse valid data to mimic real scenarios falsely.

Intra-vehicle attacks focus on the internal vehicle network, particularly the CAN Bus. DoS attacks here flood the network to halt communication between vehicle components. Spoofing sends unauthorized messages pretending to be legitimate, potentially hijacking controls like brakes. Fuzzing sends random data frames to test responses, possibly leading to faults. Falsifying attacks manipulate message contents, causing incorrect readings of vehicle status, such as speed or fuel levels. Each attack method compromises specific security goals, including availability, integrity, confidentiality, and authentication, making robust cybersecurity essential for safe vehicular operations.

**1.4.1 Challenges**

The Internet of Vehicles (IoV) is transforming the way our vehicles interact with each other and with the infrastructure around us, bringing us closer to a future of safer and more efficient transportation. However, this interconnected world also introduces a range of challenges, especially when it comes to ensuring the security of these systems. Network Intrusion Detection Systems (IDS) are meant to safeguard these networks, but as IoV technology evolves, so do the obstacles in making sure these systems remain effective.

**1.4.1.1. The Challenge of Scalability**

As more and more vehicles, roads, and devices become connected, IoV networks are growing at an incredible rate. This means that monitoring all the data being exchanged in real time becomes a huge challenge. Traditional IDS systems might struggle to keep up with the sheer volume of traffic, and they could miss crucial threats or cause delays in detecting malicious activity. The ability to scale security systems without slowing down network performance is a major hurdle for both IoV and IDS.

**1.4.1.2. Communication Protocols – A Web of Complexity**

In an IoV ecosystem, there’s no one-size-fits-all communication protocol. Different vehicles and infrastructure systems use different methods to talk to each other, like the CAN bus used within cars or LTE-V used for high-speed communication between vehicles and infrastructure. Each of these protocols has its own set of vulnerabilities. For example, the CAN bus has a simpler structure, making it an easy target for hackers, while LTE-V might have more complex needs for intrusion detection. Developing an IDS that can work across such a diverse array of communication systems adds complexity to the challenge.

**1.4.1.3. Real-Time Threat Detection**

Imagine you're driving your car, and suddenly a security threat emerges in the network whether it’s a cyberattack or an unauthorized breach. The ability to detect and respond to these threats in real time is absolutely critical. However, due to the fast-moving nature of vehicles and the high volume of data being exchanged, detecting these threats as they happen is extremely challenging. IDS systems need to be fast, efficient, and capable of identifying threats instantly. But achieving this speed without overwhelming the system or triggering false alarms is a fine line to walk.

**1.4.1.4. The Constant Evolution of Cyber Threats**

As IoV technology evolves, so too does the landscape of cyber threats. Attackers are always coming up with new methods to exploit vulnerabilities. Traditional IDS systems, particularly signature-based systems that rely on known patterns of attacks, can only catch threats they already know about. But what happens when a hacker comes up with a new, undetected method? Anomaly-based IDS are better at catching unknown threats by looking for unusual patterns, but they still have their limitations. The challenge is ensuring these systems can adapt as quickly as the threats themselves, which is no easy feat.

**1.4.1.5. Limited Resources in Vehicles**

Another major challenge in securing IoV systems is that vehicles have limited resources. Unlike large data centers or powerful servers, vehicles have restricted processing power, memory, and energy. Running an IDS on a vehicle needs to be lightweight and efficient, meaning it has to detect threats without slowing down the vehicle or draining its power. This is especially challenging when more advanced intrusion detection methods, like those based on Machine Learning, require significant computational power. Finding the right balance between effective security and limited resources is a constant struggle.

**1.4.1.6. Protecting Privacy While Ensuring Security**

With all the data flowing between vehicles and the infrastructure around them, privacy concerns are unavoidable. The information being shared, like location data, vehicle performance, and even personal driving habits, can be sensitive. Securing this data while also ensuring it’s being used to detect potential threats is tricky. Cryptographic methods can protect data, but they can also add layers of complexity to the IDS, making it harder to monitor and analyze in real time. The challenge here is to keep data secure without compromising the IDS’s ability to detect attacks.

**1.4.1.7. Building Trust Among All Parties**

IoV involves many different stakeholders—vehicle manufacturers, infrastructure providers, governments, and third-party developers—each with their own interests and security protocols. To make the IoV work, all these players need to collaborate and share data securely. But fostering trust among these different entities isn’t always easy. For example, who should take responsibility for a security breach, and how do you ensure that everyone involved follows the same security standards? Ensuring collaboration and trust between all these entities is key to making IoV systems both functional and secure.

**1.4.1.8. AI and Machine Learning – The Promise and the Problems**

Artificial Intelligence (AI) offers a lot of potential to improve IDS systems in IoV environments, especially when it comes to detecting complex, subtle threats. Machine Learning (ML) and Deep Learning could help IDS systems become more adaptable, catching new types of attacks before they cause serious damage. But AI also presents its own challenges. For example, training AI models for intrusion detection requires large, diverse datasets, which are hard to come by in the rapidly changing IoV world. And because AI systems can be tricked by adversarial attacks—where hackers manipulate data to deceive the AI—IDS using AI must be resilient to these new attack methods. The difficulty lies in developing AI-powered IDS that are both smart enough to detect threats and robust enough to avoid manipulation.

**1.4.1.9. Legal and Regulatory Hurdles**

As IoV continues to grow, it’s becoming clear that there needs to be stronger regulation around data privacy, cybersecurity, and responsibility in the event of an attack. Different regions have different laws—like GDPR in Europe—so creating IDS systems that comply with these regulations is challenging, especially when you consider the global nature of IoV networks. Additionally, if a vehicle is hacked or manipulated, determining who is legally responsible for the breach can be complicated. Developing legal frameworks that protect privacy while allowing for effective intrusion detection and response is another complex challenge.



**Figure 1.3 IoV Security challenges**

The image illustrates key security challenges faced by the Internet of Vehicles (IoV). These include issues like scalability, where handling increasing numbers of connected devices becomes difficult, and real-time detection, which is vital for identifying threats as they occur. Resource limitations refer to the constraints in computational power and memory in vehicle systems. Protocol complexity makes integration and communication between diverse systems more challenging. Evolving threats highlight the need for continuous updates to address new attack vectors. Finally, the balance between privacy and security represents the ongoing tension between protecting user data and ensuring system safety.

# **CHAPTER 2**

**LITERATURE SURVEY**

## **2. Literature Review**

The Internet of Vehicles (IoV) is a revolutionary concept that connects vehicles to each other, infrastructure, and the cloud, creating a vast network of communication. While this brings about numerous benefits like enhanced safety and efficiency, it also exposes the system to a range of cyber threats. As a result, Intrusion Detection Systems (IDS) have become essential to protect both the communication between vehicles and between vehicles and the infrastructure.

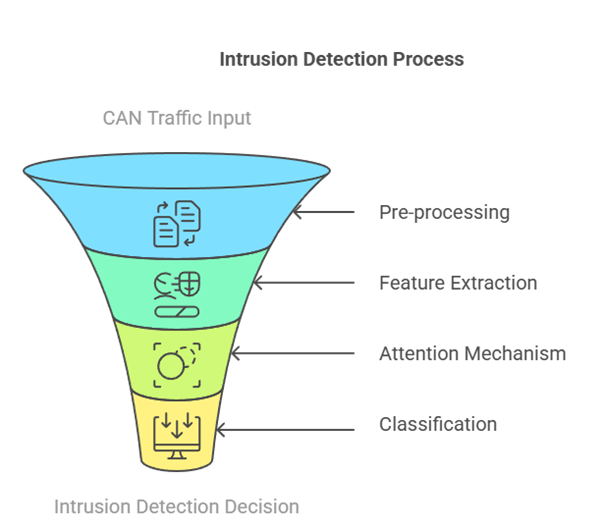
Researchers have worked with several datasets, such as CICIDS2017, NSL-KDD, CSE-CIC-IDS2018, ToN-IoT, and AWID, to model and detect different kinds of cyberattacks, including DoS, DDoS, spoofing, port scanning, phishing, and protocol exploitation. In the context of inter-vehicle systems, attacks like message falsification, impersonation, rogue updates, and eavesdropping are some of the major concerns. These attacks aim to undermine the availability, integrity, and confidentiality of the communication networks.

For intra-vehicle networks, particularly CAN buses, vulnerabilities like fuzzy attacks, gear/RPM spoofing, ID fabrication, and denial-of-service conditions are common. Researchers have applied various machine learning algorithms, such as Random Forests, Support Vector Machines (SVM), and deep learning models like LSTM and CNN, to detect anomalies and identify malicious activities within vehicle networks.

However, securing IoV comes with its own set of challenges. Some of the most pressing concerns include scalability, the need for real-time detection, constantly evolving cyber threats, limited computational resources, and the balancing act between privacy and security. While current IDS frameworks are addressing some of these issues, they still face difficulties in dealing with the complexity of protocols and the need for accurate, timely threat detection. As cyber threats grow more sophisticated, it is clear that ongoing research and development are needed to create IDS solutions that are adaptive and capable of handling the dynamic nature of IoV environments.

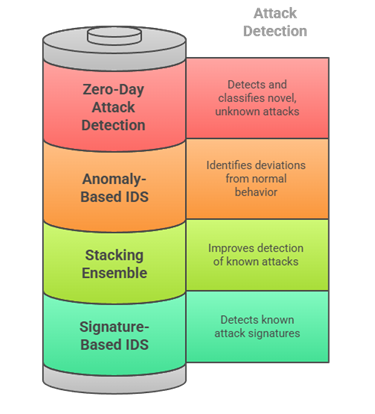
**Table2.1 Studies on Intrusion Detection Systems for IoV**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year** | **Paper** | **Methodology** | **Dataset** | **Accuracy** | **Key Focus** | **Challenges** |
| 2021 | [1] TCAN-IDS: Temporal CAN IDS using CNN | Recurrence plots + CNN | Public CAN datasets, HCRL dataset | 99.99% | High accuracy for CAN traffic; time-series analysis | Needs retraining for new IDs; limited multi-type detection |
| 2021 | [2] MTH-IDS: Multi-Tiered Hybrid IDS | Multi-tiered with BO-TPE optimization | CAN-Intrusion, CICIDS2017 | 99.88% | Detects zero-day & known attacks | Complex architecture; high runtime overhead |
| 2021 | [4] ML Technologies for Secure Vehicular Communication in IoV | ML, DL + RL | CICIDS2017, KDD CUP 99 | 99%+ | Focus on anomaly detection & privacy | High computational demands; topology dynamics |
| 2022 | [5] Security of Autonomous Vehicles in 5G IoV | Blockchain-based approach | Real-world vehicular data | Not stated | Tackles MitM, DoS, Sybil attacks | High latency; blockchain integration complexity |
| 2022 | [7] Federated AI-Enabled In-Vehicle IDS | ConvLSTM + FL + PPO client selection | Real-world IVN datasets | 95%+ | FL-based low-latency IDS | High FL overhead; complex coordination |
| 2023 | [8] FL-MAAE: IDS with FL & Memory-Augmented Autoencoder | MAAE + FL | NSL-KDD | 9.6% improvement over baseline | Unknown attack detection; data privacy | Imbalanced FL aggregation may reduce accuracy |
| 2023 | [9] IDS in Intelligent Connected Vehicles Using Weighted Self-Information | LOF with sliding window | HCRL CAN dataset | 97.42% (DoS detection) | Detects low-freq injection attacks | Limited scalability; latency from LOF |
| 2023 | [14] IDS for 5G-Enabled IoV | ML (DT, RF) on 5G vehicular traffic | NS-3 & SUMO (Simulated) | F1: 1.00 (DT), 0.98 (RF) | Detects flooding attacks | Simulation-dependent; real-world transferability |
| 2023 | [20] Enhancing IDS in IoV Through FL | FL + SMOTE + hyperparameter tuning | CICIDS2017, KDD99, UNSW-NB15 | Not specified | Privacy-preserving FL-based IDS | Class imbalance; convergence stability |
| 2023 | [21] Taxonomy on FL for IDS in IoV | Survey and taxonomy | Not applicable | Not applicable | Overview of FL approaches in IoV | Theoretical; lacks implementation |
| 2023 | [24] StatGraph: Multi-view Statistical Graph Learning for In-Vehicle IDS | Statistical graphs (TCG & CRG) + GCN | Two real in-vehicle CAN datasets | Not specified | Fine-grained intrusion detection; handles new attack types | Complexity in graph construction; resource constraints |
| 2023 | [25] Enhancing IDS Performance on Edge of Things | Feature selection + ML models | Not specified | Not specified | Improves IDS performance at the edge | Limited details on methodology and evaluation |
| 2023 | [26] Innovative IDS for High-Density Communication Networks Using AI | AI-based approach | Not specified | Not specified | Addresses IDS in high-density networks | Lack of detailed methodology and evaluation |
| 2024 | [3] Deep Transfer Learning for IDS in IoV | Transfer learning + anomaly detection | CICIDS2017, CAN | 97.2%+ | Improves generalization with less data | Model adaptation; data quality dependency |
| 2024 | [6] IDS for AVs Using Non-Tree ML Models | SVM, KNN | CICIDS2017, CAN-Intrusion | 99.87% | Detects AV network anomalies | Topology changes impact detection |
| 2024 | [12] Lifelong Learning IDS for 6G-IoV | Class-incremental + FL | Recent vehicular data | High accuracy, low false positives | Handles evolving threats | Resource limits; update complexity |
| 2024 | [13] Multi-Stage IDS via Hierarchical FL | Hierarchical FL | Not specified | Not specified | Collaborative in-vehicle IDS | Model synchronization; privacy issues |
| 2024 | [15] Hybrid IDS for IoT Devices | LSTM-GRU + tuned hyperparameters | CICIDS2017 | 95%+ | Efficient, lightweight IDS for IoT | Needs real-time efficiency |
| 2024 | [16] Multi-Stage IDS Framework for IoT | Layered detection architecture | Not specified | Not specified | Layered IDS enhances detection | Scalability and integration challenges |
| 2024 | [17] Resilient Adaptive Event Trigger IDS | Event-triggered control model | Not specified | Not specified | DoS and deception attack defense | Ensuring timely, accurate response |
| 2024 | [18] AI-Based IDS – A Survey | Literature review | Not applicable | Not applicable | Overview of AI use in IDS | Informational; lacks testing |
| 2024 | [19] Dynamic Hierarchical IDS on Edge | Hierarchical detection on edge platform | Not specified | Not specified | Enhances speed and detection via edge nodes | Latency vs. complexity balance |
| 2024 | [22] Interpretable IDS with Discrete Optimization Learning | Discrete optimization for proactive IDS | Not specified | Not specified | Proactive and interpretable models | Optimization complexity |
| 2024 | [27] Zero-X: Blockchain-Enabled Open-Set FL Framework for Zero-Day Attack Detection in IoV | Deep neural networks + Open-Set Recognition + Blockchain-based FL | Two recent network traffic datasets | High detection rate; low false positive rate | Detects zero-day and N-day attacks; privacy-preserving FL | Complexity in integrating blockchain with FL; resource constraints |
| 2024 | [28] Homomorphic Encryption-Enabled FL for Privacy-Preserving IDS in Resource-Constrained IoV Networks | Homomorphic encryption + FL | Simulated datasets | Performance gap < 0.8% compared to non-encrypted FL | Privacy-preserving IDS for resource-constrained IoV | Computational overhead due to encryption |
| 2024 | [29] Deep Learning Enabled IDS for Industrial IoT Environment | Deep learning models | Not specified | Not specified | Enhances IDS in Industrial IoT settings | Lack of specific dataset and evaluation metrics |
| 2024 | [30] Dynamic Hierarchical IDS for IoV on Edge Computing Platform | Hierarchical clustering + edge computing | Simulated datasets | Not specified | Protects infrastructure and passengers in IoV | Scalability and deployment challenges |
| 2025 | [10] Modular Zero-Day Botnet Detection Using Isolation Forests + PSO | Meta-ensemble model | Botnet dataset | 92.8% (known), 77.3% (zero-day) | Detects evolving botnet attacks | High edge computation requirements |
| 2025 | [11] ATHENA: Physical Feature-Based IDS | LSTM + clustering; vehicle-cloud hybrid | ROAD dataset | Not specified | Uses physical layer behavior for IDS | Accurate modeling needed; latency issues |
| 2025 | [23] Optimized Fick’s Law Graph Point IDS | Convolutional sparse trans-net model | Not specified | Not specified | Industrial IoV threat detection | High computational load; futuristic model |
| 2025 | [31] Evolutionary LightGBM-Based IDS for IoT Networks | LightGBM with evolutionary algorithms | Not specified | Not specified | Enhances IDS in IoT networks | Lack of specific dataset and evaluation metrics |



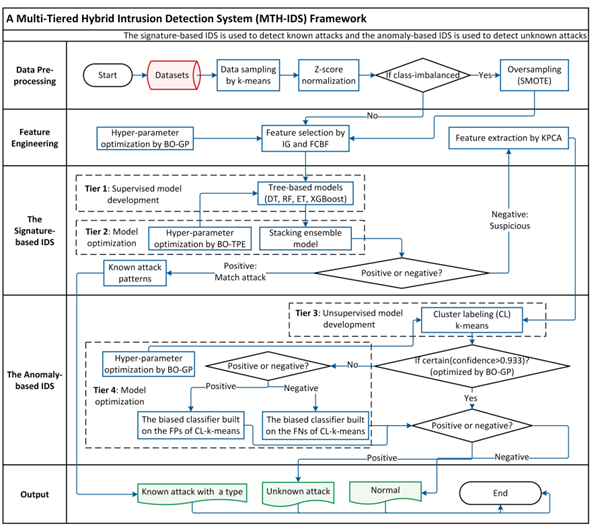
**Figure 2.1 Flowchart of Intrusion Detection Process**

The figure 2.1 illustrates a structured intrusion detection process for vehicular networks using CAN (Controller Area Network) traffic as input. The process begins with a **pre-processing** stage, where raw CAN data is cleaned and transformed into a suitable format for analysis. This step ensures the removal of irrelevant or noisy data that may interfere with detection accuracy. Next, in the **feature extraction** phase, meaningful characteristics and patterns are derived from the processed data, helping the system to distinguish between normal and potentially malicious behavior. The **attention mechanism** is then applied to highlight and prioritize the most relevant features, allowing the model to focus on critical aspects of the data. Finally, in the **classification** stage, a machine learning or deep learning model is used to analyze the extracted and refined features, ultimately producing an **intrusion detection decision**. This layered approach enables the system to effectively detect and respond to cyber threats in real-time vehicular environments.



**Figure 2.2 Flowchart Of Different Detection In IDS**

The figure represents the **attack detection components** in a layered intrusion detection system, showcasing four key strategies used to identify various types of cyber threats. At the top is **Zero-Day Attack Detection**, which focuses on identifying novel or previously unseen attacks that have no known signature, making it crucial for protecting against emerging threats. Below that, the **Anomaly-Based IDS** detects deviations from established normal behavior patterns, enabling the system to flag suspicious activities that may indicate a security breach. The **Stacking Ensemble** layer combines multiple detection models to enhance accuracy, particularly in recognizing known attack types by leveraging the strengths of individual models. Finally, the **Signature-Based IDS** relies on pre-defined patterns or signatures to accurately detect well-known attacks. Together, these layers provide a comprehensive, multi-dimensional defense mechanism against a wide range of cyber threats in vehicular networks



**Figure 2.3 Framework MTH**

The diagram illustrates the architecture of the **Multi-Tiered Hybrid Intrusion Detection System (MTH-IDS)**, designed to detect both known and unknown cyber-attacks in vehicular networks. It is divided into four main stages: **Data Pre-processing**, **Feature Engineering**, **Signature-based IDS**, and **Anomaly-based IDS**.

The process begins with **data pre-processing**, where datasets are sampled using k-means and normalized with Z-score. If the dataset is imbalanced, oversampling is applied using SMOTE. Next, in the **feature engineering** phase, relevant features are selected using IG and FCBF methods, or extracted via KPCA for dimensionality reduction, with hyperparameter tuning done through Bayesian Optimization using Gaussian Process (BO-GP).

The **signature-based IDS** consists of two tiers. **Tier 1** involves building supervised models (like Decision Trees, Random Forests, Extra Trees, and XGBoost) trained on known attack patterns. **Tier 2** improves accuracy by applying Bayesian Optimization with Tree-structured Parzen Estimator (BO-TPE) to tune a stacking ensemble model. If the output is positive (i.e., a known attack), the result is reported. If it's negative, the input is treated as suspicious and passed to the anomaly-based IDS.

In the **anomaly-based IDS**, **Tier 3** uses unsupervised clustering (CL-k-means) to group and label data, while **Tier 4** applies further hyperparameter optimization and classification. Depending on confidence levels, data is evaluated for possible classification. If confidently classified as normal or attack, it is labeled accordingly; otherwise, it is flagged as an unknown attack.

Ultimately, this multi-layered approach yields three possible outcomes: a known attack with its type, an unknown attack, or normal traffic—offering a robust and comprehensive IDS framework for the Internet of Vehicles.

# **CHAPTER 3**

**DATASET OVERVIEW**

## **3. Dataset Overview**

The dataset used in this study consists of the CICIDS2017 [32] and CICIDS2018 [33] datasets, both of which are publicly available from the Canadian Institute for Cybersecurity. These datasets contain real-world traffic data collected from simulated network environments with a wide variety of benign and attack traffic, making them suitable for training and evaluating Intrusion Detection Systems (IDS) for the Internet of Vehicles (IoV).

**CICIDS2017** includes network traffic data recorded over multiple days with several attack types, including DDoS, Brute Force, and PortScan. **CICIDS2018** offers a similar set of traffic data with additional attack types like Infiltration and Botnet. These datasets provide features such as flow-based network traffic characteristics, including packet size, duration, protocol type, and byte counts, which serve as input features for the IDS models.

## **3.1 CICIDS2017 Dataset**

The CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System 2017) dataset consists of labeled network traffic data collected over five days. It includes a mix of normal and malicious traffic, covering various attack types such as:

* Brute Force Attacks (SSH, FTP)
* DoS (Denial of Service) Attacks (Slowloris, Hulk, GoldenEye, Heartbleed)
* Web Attacks (SQL Injection, XSS)
* Botnet Activities
* DDoS Attacks
* Infiltration Attacks
* Port Scanning

Features:

* The dataset includes 80+ extracted features from network traffic, derived using CICFlowMeter.
* Features include packet-based statistics, flow-based characteristics, header information, and payload analysis.

**Table 3.1 All Features in CICIDS2017**

|  |  |  |
| --- | --- | --- |
| **No.** | **Feature Name** | **Detail** |
| 1 | Flow ID | Unique identifier for flow |
| 2 | Source IP | IP address of sender |
| 3 | Source Port | Port number of sender |
| 4 | Destination IP | IP address of receiver |
| 5 | Destination Port | Port number of receiver |
| 6 | Protocol | Protocol type (TCP/UDP/ICMP) |
| 7 | Timestamp | Time of first packet |
| 8 | Flow Duration | Duration of the flow |
| 9 | Total Fwd Packets | Number of packets forward |
| 10 | Total Backward Packets | Number of packets backward |
| 11 | Total Length of Fwd Packets | Total bytes sent forward |
| 12 | Total Length of Bwd Packets | Total bytes sent backward |
| 13 | Fwd Packet Length Max | Max size of forward packet |
| 14 | Fwd Packet Length Min | Min size of forward packet |
| 15 | Fwd Packet Length Mean | Avg size of forward packets |
| 16 | Fwd Packet Length Std | Std deviation of fwd packets |
| 17 | Bwd Packet Length Max | Max size of backward packet |
| 18 | Bwd Packet Length Min | Min size of backward packet |
| 19 | Bwd Packet Length Mean | Avg size of backward packets |
| 20 | Bwd Packet Length Std | Std deviation of bwd packets |
| 21 | Flow Bytes/s | Flow byte rate (bytes/sec) |
| 22 | Flow Packets/s | Flow packet rate (packets/sec) |
| 23 | Flow IAT Mean | Avg inter-arrival time flow |
| 24 | Flow IAT Std | Std deviation of IAT |
| 25 | Flow IAT Max | Max inter-arrival time flow |
| 26 | Flow IAT Min | Min inter-arrival time flow |
| 27 | Fwd IAT Total | Total fwd inter-arrival time |
| 28 | Fwd IAT Mean | Avg fwd IAT |
| 29 | Fwd IAT Std | Std deviation of fwd IAT |
| 30 | Fwd IAT Max | Max fwd IAT |
| 31 | Fwd IAT Min | Min fwd IAT |
| 32 | Bwd IAT Total | Total bwd inter-arrival time |
| 33 | Bwd IAT Mean | Avg bwd IAT |
| 34 | Bwd IAT Std | Std deviation of bwd IAT |
| 35 | Bwd IAT Max | Max bwd IAT |
| 36 | Bwd IAT Min | Min bwd IAT |
| 37 | Fwd PSH Flags | Number of PSH flags fwd |
| 38 | Bwd PSH Flags | Number of PSH flags bwd |
| 39 | Fwd URG Flags | Number of URG flags fwd |
| 40 | Bwd URG Flags | Number of URG flags bwd |
| 41 | Fwd Header Length | Forward header size total |
| 42 | Bwd Header Length | Backward header size total |
| 43 | Fwd Packets/s | Forward packets per second |
| 44 | Bwd Packets/s | Backward packets per second |
| 45 | Min Packet Length | Minimum packet size |
| 46 | Max Packet Length | Maximum packet size |
| 47 | Packet Length Mean | Mean packet size |
| 48 | Packet Length Std | Std deviation of packet size |
| 49 | Packet Length Variance | Variance of packet size |
| 50 | FIN Flag Count | Number of FIN flags |
| 51 | SYN Flag Count | Number of SYN flags |
| 52 | RST Flag Count | Number of RST flags |
| 53 | PSH Flag Count | Number of PSH flags |
| 54 | ACK Flag Count | Number of ACK flags |
| 55 | URG Flag Count | Number of URG flags |
| 56 | CWE Flag Count | Number of CWE flags |
| 57 | ECE Flag Count | Number of ECE flags |
| 58 | Down/Up Ratio | Download/Upload packet ratio |
| 59 | Average Packet Size | Average size of packets |
| 60 | Fwd Segment Size Avg | Average segment size fwd |
| 61 | Bwd Segment Size Avg | Average segment size bwd |
| 62 | Fwd Bytes/Bulk Avg | Avg bytes per bulk fwd |
| 63 | Fwd Packet/Bulk Avg | Avg packets per bulk fwd |
| 64 | Fwd Bulk Rate Avg | Avg bulk rate fwd |
| 65 | Bwd Bytes/Bulk Avg | Avg bytes per bulk bwd |
| 66 | Bwd Packet/Bulk Avg | Avg packets per bulk bwd |
| 67 | Bwd Bulk Rate Avg | Avg bulk rate bwd |
| 68 | Subflow Fwd Packets | Packets in subflow fwd |
| 69 | Subflow Fwd Bytes | Bytes in subflow fwd |
| 70 | Subflow Bwd Packets | Packets in subflow bwd |
| 71 | Subflow Bwd Bytes | Bytes in subflow bwd |
| 72 | Init\_Win\_bytes\_forward | Initial window size fwd |
| 73 | Init\_Win\_bytes\_backward | Initial window size bwd |
| 74 | act\_data\_pkt\_fwd | Number of data pkts fwd |
| 75 | min\_seg\_size\_forward | Min segment size fwd |
| 76 | Active Mean | Mean active time flow |
| 77 | Active Std | Std deviation active time |
| 78 | Active Max | Max active time |
| 79 | Active Min | Min active time |
| 80 | Idle Mean | Mean idle time |
| 81 | Idle Std | Std deviation idle time |
| 82 | Idle Max | Max idle time |
| 83 | Idle Min | Min idle time |
| 84 | Label (Benign or Attack Type) | Target class (output label) |

File Formats:

* PCAP files (raw network traffic)
* CSV files (preprocessed network flow features)

## **3.2 CICIDS2018 Dataset**

The CICIDS2018 dataset is an improved version of CICIDS2017, with a more diverse attack landscape and better feature extraction. It captures real-world cyber threats, including [32]:

* Brute Force Attacks
* Botnet Attacks
* DoS/DDoS Attacks
* Web Exploits
* Infiltration Attacks
* Cryptojacking (Cryptocurrency Mining)
* Man-in-the-Middle (MITM) Attacks

**Features:**

* The dataset contains more than 80 features, similar to CICIDS2017, but with refinements and additional attributes to better capture modern cyber threats.
* These features help in developing machine learning-based intrusion detection systems (IDS).

**Table 3.2 All Features in CICIDS2018**

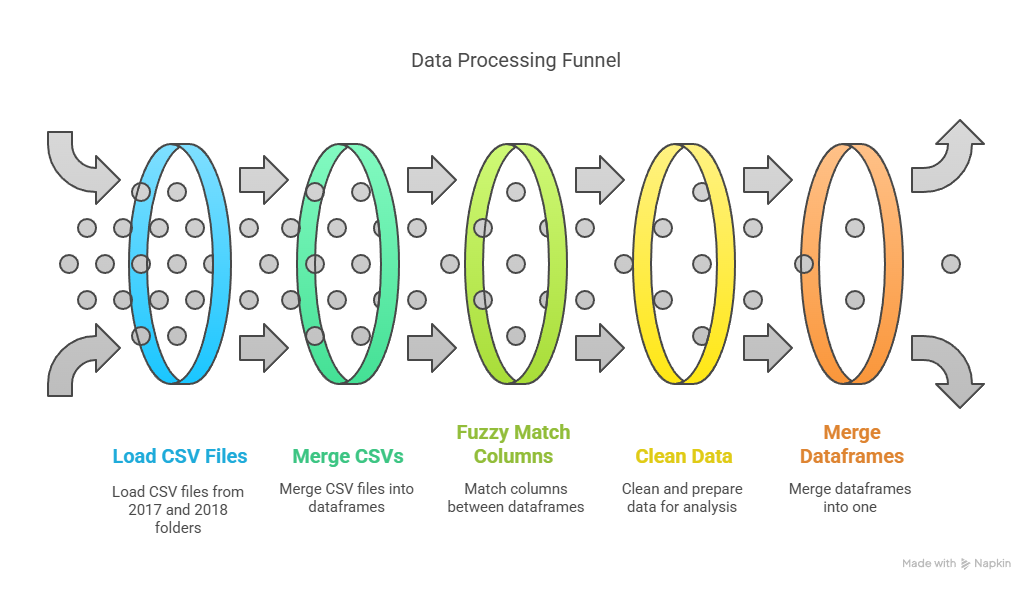
|  |  |  |
| --- | --- | --- |
| **No.** | **Feature Name** | **Detail** |
| 1 | Destination Port | Port number of receiver |
| 2 | Flow Duration | Duration of the flow |
| 3 | Total Fwd Packets | Number of packets forward |
| 4 | Total Backward Packets | Number of packets backward |
| 5 | Total Length of Fwd Packets | Total bytes sent forward |
| 6 | Total Length of Bwd Packets | Total bytes sent backward |
| 7 | Fwd Packet Length Max | Max size of forward packet |
| 8 | Fwd Packet Length Min | Min size of forward packet |
| 9 | Fwd Packet Length Mean | Avg size of forward packets |
| 10 | Fwd Packet Length Std | Std deviation of fwd packets |
| 11 | Bwd Packet Length Max | Max size of backward packet |
| 12 | Bwd Packet Length Min | Min size of backward packet |
| 13 | Bwd Packet Length Mean | Avg size of backward packets |
| 14 | Bwd Packet Length Std | Std deviation of bwd packets |
| 15 | Flow Bytes/s | Flow byte rate (bytes/sec) |
| 16 | Flow Packets/s | Flow packet rate (packets/sec) |
| 17 | Flow IAT Mean | Avg inter-arrival time flow |
| 18 | Flow IAT Std | Std deviation of IAT |
| 19 | Flow IAT Max | Max inter-arrival time flow |
| 20 | Flow IAT Min | Min inter-arrival time flow |
| 21 | Fwd IAT Total | Total fwd inter-arrival time |
| 22 | Fwd IAT Mean | Avg fwd IAT |
| 23 | Fwd IAT Std | Std deviation of fwd IAT |
| 24 | Fwd IAT Max | Max fwd IAT |
| 25 | Fwd IAT Min | Min fwd IAT |
| 26 | Bwd IAT Total | Total bwd inter-arrival time |
| 27 | Bwd IAT Mean | Avg bwd IAT |
| 28 | Bwd IAT Std | Std deviation of bwd IAT |
| 29 | Bwd IAT Max | Max bwd IAT |
| 30 | Bwd IAT Min | Min bwd IAT |
| 31 | Fwd PSH Flags | Number of PSH flags fwd |
| 32 | Bwd PSH Flags | Number of PSH flags bwd |
| 33 | Fwd URG Flags | Number of URG flags fwd |
| 34 | Bwd URG Flags | Number of URG flags bwd |
| 35 | Fwd Header Length | Forward header size total |
| 36 | Bwd Header Length | Backward header size total |
| 37 | Fwd Packets/s | Forward packets per second |
| 38 | Bwd Packets/s | Backward packets per second |
| 39 | Min Packet Length | Minimum packet size |
| 40 | Max Packet Length | Maximum packet size |
| 41 | Packet Length Mean | Mean packet size |
| 42 | Packet Length Std | Std deviation of packet size |
| 43 | Packet Length Variance | Variance of packet size |
| 44 | FIN Flag Count | Number of FIN flags |
| 45 | SYN Flag Count | Number of SYN flags |
| 46 | RST Flag Count | Number of RST flags |
| 47 | PSH Flag Count | Number of PSH flags |
| 48 | ACK Flag Count | Number of ACK flags |
| 49 | URG Flag Count | Number of URG flags |
| 50 | CWE Flag Count | Number of CWE flags |
| 51 | ECE Flag Count | Number of ECE flags |
| 52 | Down/Up Ratio | Download/Upload packet ratio |
| 53 | Average Packet Size | Average size of packets |
| 54 | Fwd Segment Size Avg | Average segment size fwd |
| 55 | Bwd Segment Size Avg | Average segment size bwd |
| 56 | Fwd Bytes/Bulk Avg | Avg bytes per bulk fwd |
| 57 | Fwd Packet/Bulk Avg | Avg packets per bulk fwd |
| 58 | Fwd Bulk Rate Avg | Avg bulk rate fwd |
| 59 | Bwd Bytes/Bulk Avg | Avg bytes per bulk bwd |
| 60 | Bwd Packet/Bulk Avg | Avg packets per bulk bwd |
| 61 | Bwd Bulk Rate Avg | Avg bulk rate bwd |
| 62 | Subflow Fwd Packets | Packets in subflow fwd |
| 63 | Subflow Fwd Bytes | Bytes in subflow fwd |
| 64 | Subflow Bwd Packets | Packets in subflow bwd |
| 65 | Subflow Bwd Bytes | Bytes in subflow bwd |
| 66 | Init\_Win\_bytes\_forward | Initial window size fwd |
| 67 | Init\_Win\_bytes\_backward | Initial window size bwd |
| 68 | act\_data\_pkt\_fwd | Number of data pkts fwd |
| 69 | min\_seg\_size\_forward | Min segment size fwd |
| 70 | Active Mean | Mean active time flow |
| 71 | Active Std | Std deviation active time |
| 72 | Active Max | Max active time |
| 73 | Active Min | Min active time |
| 74 | Idle Mean | Mean idle time |
| 75 | Idle Std | Std deviation idle time |
| 76 | Idle Max | Max idle time |
| 77 | Idle Min | Min idle time |
| 78 | Label (Benign or Attack Type) | Target class (output label) |

File Formats:

* PCAP files (raw network traffic)
* CSV files (preprocessed network flow features)

## **3.3 Data Processing**

To create a unified dataset, we applied **fuzzy matching** techniques to align the features from both datasets, ensuring compatibility between them. Two primary datasets were generated:

1. **cicidsmerge\_binary.csv**: This dataset is used for binary classification tasks, with labels indicating whether the traffic is benign or an attack.
2. **cicidsmerge\_multiclass.csv**: This dataset is used for multiclass classification tasks, with each attack type classified into specific categories (e.g., DDoS, Brute Force, PortScan).

**Figure3.1 Data Processing Funnel**

The data underwent several preprocessing steps, including **label encoding** for categorical features, **null value imputation**, and **normalization** using MinMaxScaler. Additionally, the data was reshaped into a 3D format suitable for LSTM input, where each sample corresponds to a time step sequence of features.

The Figure 3.1 presents a **Data Processing Funnel** used to organize and combine data from multiple CSV files, particularly those from the CICIDS2017 and CICIDS2018 datasets. It starts with loading CSV files from both years into separate DataFrames. These files are then merged individually to form two combined datasets—one for each year. The next phase focuses on aligning the column names using fuzzy matching techniques to account for slight differences in naming conventions between the datasets. After matching the columns, the data is cleaned by addressing missing values, converting data types, and standardizing the format. In the final step, the two cleaned datasets are merged into a single DataFrame, creating a unified dataset ready for analysis or machine learning tasks. This funnel offers a systematic approach to preparing large-scale intrusion detection data for further use.

# **CHAPTER 4**

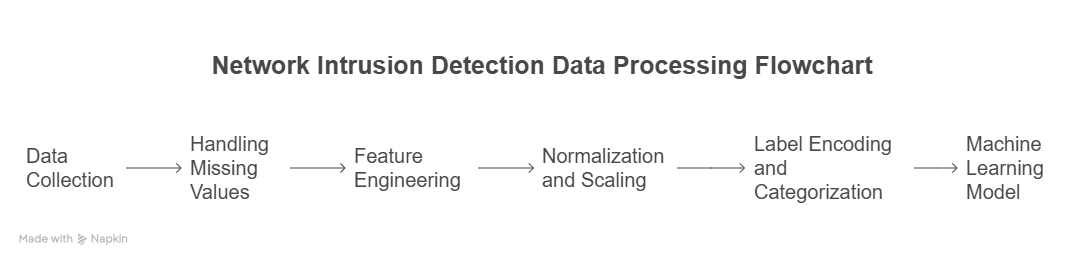
**METHODOLOGY**

## **4. Methodology of Project**

This study adopts a comprehensive and systematic approach for building an **Intrusion Detection System (IDS)** tailored to the **Internet of Vehicles (IoV)**. The methodology focuses on leveraging machine learning techniques, particularly ensemble learning models, to identify and classify potential intrusions in vehicular networks. The approach involves several stages, from data acquisition and preprocessing to model selection, training, evaluation, and deployment.

## **4.1. Data Acquisition and Preprocessing**

The dataset used for this study is based on the **CICIDS2017** and **CICIDS2018** datasets, which are among the most widely recognized datasets for network intrusion detection. These datasets contain labeled network traffic data, representing both benign and malicious activities. They are generated by monitoring the network traffic of a simulated vehicular environment, which includes both normal behavior and attacks like **Brute Force**, **DDoS**, **Heartbleed**, **Web Attacks**, and others.

* **Handling Missing Values**: Missing data points are an inevitable issue in network traffic datasets. In this step, missing values are handled using techniques like **mean imputation** for numerical data and **mode imputation** for categorical features. This ensures that the dataset remains complete and ready for analysis.
* **Feature Engineering**: Feature engineering is essential for improving the predictive power of machine learning models. The raw network traffic data includes a variety of features such as **IP addresses**, **protocol types**, **ports**, **packet sizes**, and others. These raw features are processed and transformed into meaningful attributes. For example, **time-based features** such as packet arrival time, request-response time, and session duration are computed to capture temporal patterns in the data. **Statistical features** like average packet size, mean inter-arrival times, and packet count per flow are also included, as they are vital in identifying attack patterns.
* **Normalization and Scaling**: Since the features in network traffic data vary significantly in scale, **Min-Max scaling** or **Z-score normalization** is applied to bring all features to a comparable range. This helps machine learning models, especially distance-based algorithms, to converge quickly and improve performance.
* **Label Encoding and Categorization**: The dataset includes both benign and attack data. To make the data suitable for machine learning models, the attack labels are converted to numerical categories using **label encoding**. This step ensures that the model can easily process categorical data and classify network flows accordingly.

**Figure 4.1 Network Intrusion Detection Data Processing**

## **4.2. Model Selection and Design**

The heart of the IDS lies in the choice of machine learning models. In this study, two powerful algorithms are used:

**4.2.1 Random Forest (RF)**

Random Forest is an ensemble learning method designed to improve the performance and stability of decision trees. Instead of relying on a single decision tree, Random Forest constructs a large collection of trees during training. Each tree is trained on a random subset of the original dataset, and predictions are made by aggregating the outputs of all trees, typically using majority voting for classification or averaging for regression tasks.

One of the major advantages of Random Forest is its ability to reduce overfitting, a common weakness in individual decision trees. By training multiple trees on different random samples and random subsets of features, Random Forest ensures diversity among the models, leading to better generalization to unseen data.

Key features of Random Forest include:

* **Bagging (Bootstrap Aggregating):** Multiple trees are trained on different random samples drawn with replacement from the dataset.
* **Feature Randomness:** At each decision split, a random subset of features is considered, increasing model robustness.
* **Handling Missing Data and Imbalance:** Random Forest performs well even when the dataset has missing values or unbalanced class distributions.
* **Feature Importance:** It provides estimates of feature importance, which can guide feature selection and interpretation.

**Random Forest Calculation**:

Each tree in the forest is trained on a subset of the dataset, and predictions are made using majority voting for classification or averaging for regression.

Where:

* is the final prediction from the ensemble.
* N is the number of decision trees in the forest.
* is the prediction of the -*th* tree.

In security applications such as Intrusion Detection Systems (IDS) for the Internet of Vehicles (IoV), Random Forest is valuable because of its high accuracy, ability to handle noisy datasets, and interpretability.

## **4.2.2 Long Short-Term Memory (LSTM)**

Long Short-Term Memory networks (LSTMs) are a special type of Recurrent Neural Network (RNN) designed to model sequential and time-dependent data. Unlike basic RNNs, which often suffer from vanishing or exploding gradients during training, LSTM networks use a unique architecture with memory cells and gating mechanisms to maintain information over long sequences.

An LSTM cell includes three gates:

* **Forget Gate:** Decides what information from the previous cell state should be forgotten.
* **Input Gate:** Controls how much new information should be added to the cell state.
* **Output Gate:** Determines what information from the current cell state should be output to the next layer.

**Forget Gate ()**:  
The forget gate decides which information from previous states should be discarded. It outputs a value between 0 and 1, determining how much of the previous state to retain.

Where:

* is the output of the forget gate.
* is the sigmoid activation function:
* is the weight matrix for the forget gate.
* is the previous hidden state.
* is the current input (network traffic features at time .
* is the bias term.

**Input Gate ():**

The input gate controls how much new information should be added to the cell state. It combines the previous hidden state and the current input, passes it through a sigmoid function, and then applies the tanh function to generate candidate values

Where:

* is the input gate output.
* is the candidate cell state.
* are the weight matrices.
* are the bias terms.

**Cell State Update ()**:

The cell state is updated using the forget and input gates. It combines the previous cell state and the candidate cell state

Where:

* is the updated cell state.
* is the forget gate's output.
* is the input gate's output.
* is the candidate cell state.

**Output Gate ):**

The output gate decides what information from the cell state will be passed to the next hidden state, which will be used for the next time step and for classification.

Where:

* is the output gate's output.
* is the new hidden state.

Advantages of LSTM include:

* **Capturing Long-Term Dependencies:** LSTM can remember important information for long periods, making it suitable for tasks where the sequence of data matters.
* **Adaptability to Sequential Data:** LSTM models excel in areas like time-series prediction, language modeling, and anomaly detection.
* **Handling Variable-Length Sequences:** They can process sequences of different lengths without the need for fixed-size input.

In the context of Intrusion Detection Systems for IoV networks, LSTM can identify patterns in communication sequences between vehicles and infrastructure. It detects abnormal behaviors over time, enabling dynamic and intelligent intrusion detection.

**Table 4.1 Difference between RF and LSTM**

| **Feature** | **Random Forest (RF)** | **Long Short-Term Memory (LSTM)** |
| --- | --- | --- |
| Learning Type | Ensemble of Decision Trees | Recurrent Neural Network (RNN) |
| Best for | Structured/tabular data | Sequential/time-dependent data |
| Handling Overfitting | Yes, via bagging | Needs careful tuning |
| Strength | High accuracy, feature ranking | Long-term memory, pattern detection |
| Common Application | Static data analysis, classification | Dynamic prediction, anomaly detection |

## **4.3. Training And Evaluation Of Models**

Once the dataset is preprocessed, the models are trained using a combination of **train-test splits** and **cross-validation** to ensure the generalization of the models. **Grid Search** and **Random Search** are applied to perform **hyperparameter tuning**, allowing for optimal model configuration. The model training process involves the following:

* **Random Forest**: The Random Forest model is trained on the processed features, and the number of decision trees (n\_estimators) and the maximum depth of each tree are tuned for optimal performance.
* **LSTM**: The LSTM model is trained using sequential data, with tuning of hyperparameters like **learning rate**, **batch size**, **epochs**, and **hidden units** in the LSTM layers to prevent overfitting and ensure convergence.
* **Ensemble Models**: The OCED ensemble model combines the predictions of multiple base models. A weighted majority voting mechanism is used to make the final classification decision based on the output from each individual model.

The performance of the models is evaluated using various metrics, including:

* **Accuracy**: The proportion of correct predictions made by the model compared to the total number of predictions. A high accuracy indicates that the model is able to classify benign and malicious traffic accurately.
* **Precision, Recall, and F1-Score**: These metrics are crucial for assessing the model’s performance, especially in an imbalanced dataset like network traffic, where malicious traffic is usually much less than benign traffic. **Precision** measures the proportion of true positives out of all predicted positives, **Recall** assesses the proportion of true positives out of all actual positives, and **F1-Score** provides a balance between Precision and Recall.
* **Confusion Matrix**: A confusion matrix is used to visualize the performance of the classification model, showing the number of true positives, true negatives, false positives, and false negatives.

**4.3.1 Training the Random Forest Model for binary Classification**

The prepared dataset, containing binary labels (Benign or Attack), was used to train a Random Forest classifier. Before model training, several important preprocessing steps were applied to ensure data quality and optimize model performance.[13]

First, any rows containing missing label values were removed from the dataset. The feature set (X) and label set (y) were then separated. Since the dataset contained some non-numeric values and infinite values, all feature columns were converted to numeric types, with invalid entries replaced by NaN values and subsequently dropped. This ensured a clean and fully numeric feature matrix for training.

Label encoding was performed to convert the categorical label values into binary numeric format, where 'Benign' was mapped to 0 and 'Attack' was mapped to 1. Feature scaling was applied using Min-Max normalization to ensure that all feature values fell within the range [0, 1], which helps improve the performance of distance-based machine learning algorithms.

The dataset was then split into training and testing subsets using an 80-20 ratio. Stratification was applied during the split to maintain the same class distribution across both subsets.

The Random Forest classifier was trained with the following hyperparameters:

* **Number of Estimators:** 100 trees
* **Maximum Depth:** 20
* **Maximum Features:** Square root of total features at each split
* **Random State:** 42 (to ensure reproducibility)
* **Parallel Jobs:** All available processors (n\_jobs = -1) to accelerate training

After training, the model was ready to be evaluated on unseen data.

**Model Evaluation:**

The performance of the trained Random Forest model was assessed using the test dataset. The key evaluation metrics included:

* **Accuracy:** Measures the proportion of correctly predicted instances over the total instances.
* **F1 Score:** The harmonic mean of precision and recall, providing a balance between false positives and false negatives.
* **Classification Report:** Provides detailed performance results, including precision, recall, and F1-score for each class.
* **Confusion Matrix:** A tabular summary showing correct and incorrect predictions broken down by class.

The model achieved a high accuracy and F1-score on the test set, indicating effective learning and generalization. The confusion matrix was visualized using a heatmap to better understand the distribution of true positives, false positives, true negatives, and false negatives across the two classes (Benign and Attack).

This evaluation confirmed that the Random Forest classifier is a strong candidate for intrusion detection tasks in Internet of Vehicles (IoV) environments, offering both high performance and interpretability.

**4.3.2 Training the Random Forest Model for Multiclass Classification**

To develop a multiclass intrusion detection model, the CICIDS2017 and CICIDS2018 merged dataset with fuzzy labeling was used. Each instance in the dataset is labeled with a specific attack type or benign behavior, requiring the model to distinguish between multiple classes rather than just binary outcomes.

Prior to model training, several preprocessing steps were performed:

* **Data Cleaning:** Any rows containing missing values were removed to ensure consistency in training.
* **Feature and Label Separation:** The dataset was divided into feature matrix X and label vector y.
* **Numeric Conversion:** All features were coerced into numeric format, and rows with non-numeric or infinite values were discarded.
* **Label Encoding:** Since the labels were categorical (different attack types), they were encoded into integer values using label encoding.
* **Normalization:** Min-Max scaling was applied to the features, ensuring all values fall within the range [0, 1] to improve model learning.

After preprocessing, the dataset was split into training and testing sets using an 80-20 ratio, ensuring stratification to maintain balanced class distribution in both sets.

A Random Forest Classifier was then trained with the following configuration:

* **Number of Trees:** 100
* **Random State:** 42 (for reproducibility)
* **Parallelization:** Training was accelerated using all available CPU cores (n\_jobs=-1).

This setup enabled efficient handling of the multiclass nature of the dataset, allowing the model to build decision boundaries for multiple attack categories.

**Model Evaluation**

The performance of the trained Random Forest model was evaluated using several standard metrics:

* **Accuracy:** The overall proportion of correctly classified instances across all classes.
* **Classification Report:** A detailed report showing precision, recall, and F1-score for each class individually. This helps assess how well the model performs for each specific attack type and benign traffic.
* **Confusion Matrix:** A matrix representation of actual versus predicted classes, offering insights into which types of attacks are more frequently misclassified.

The confusion matrix was visualized using a heatmap for better interpretability. High diagonal values in the confusion matrix indicate correct classifications, while off-diagonal values highlight misclassifications.

Despite the complexity introduced by multiple classes, the Random Forest model achieved a strong overall accuracy and maintained high precision and recall values across most classes. These results confirm that ensemble methods like Random Forest are highly effective in handling multiclass intrusion detection tasks in Internet of Vehicles (IoV) environments.

**4.3.3 Training the LSTM Model for Multiclass Classification**

For sequential modeling of multiclass intrusion data, a Long Short-Term Memory (LSTM) network was developed. LSTM networks are particularly suited for capturing time-dependent or sequential patterns within data, making them effective for analyzing network traffic behaviors.

Prior to training, the dataset underwent several preprocessing steps:

* **Data Cleaning:** Infinite and missing values were identified and removed to ensure consistency.
* **Feature and Label Separation:** Features were separated from their corresponding labels.
* **Numeric Conversion:** All feature columns were converted into numeric format, with errors coerced and invalid rows dropped.
* **Label Encoding:** Class labels were encoded into integer representations suitable for multiclass classification.
* **Normalization:** Features were scaled using Min-Max scaling to fit within the range [0, 1].
* **Reshaping for LSTM:** As LSTM networks require three-dimensional input (samples, time steps, features), the feature matrix was reshaped accordingly.[13]

After preprocessing, the dataset was divided into training and testing subsets using an 80-20 split with stratification to maintain class balance.

The LSTM model was designed with the following architecture:

* **Input Layer:** Accepts the normalized and reshaped feature sequences.
* **LSTM Layer:** A recurrent layer with 64 units, responsible for learning temporal patterns.
* **Dropout Layer:** Dropout of 30% was added to prevent overfitting by randomly deactivating neurons during training.
* **Dense Layer:** A fully connected layer with 64 neurons and ReLU activation to introduce non-linearity.
* **Output Layer:** A Dense layer with a number of neurons equal to the number of classes, using Softmax activation to output class probabilities.

The model was compiled using the Adam optimizer and sparse categorical cross-entropy loss, as the task involved multiclass label prediction. The model was trained for 10 epochs with a batch size of 128, utilizing 20% of the training data for validation.

**Model Evaluation**

The performance of the LSTM model was monitored throughout training by tracking both accuracy and loss for training and validation datasets. After completion, the model was evaluated on the unseen test set.

Evaluation metrics included:

* **Classification Report:** Provides precision, recall, and F1-scores for each attack class individually, helping to identify strengths and weaknesses across different types of intrusions.
* **Confusion Matrix:** A visual representation was created to illustrate how frequently each class was correctly or incorrectly predicted.

The model demonstrated strong performance across multiple classes, achieving high accuracy levels while maintaining generalization as seen through close training and validation scores. The confusion matrix revealed that while most classes were well-separated, a few attack types exhibited minor misclassification, which is typical in multiclass IDS tasks due to overlapping behaviors between certain attacks.

Additionally, training curves for accuracy and loss were plotted to visualize model learning dynamics over epochs, confirming stable convergence without significant overfitting.

**4.3.4 Training the LSTM Model for Binary Classification**

In this phase, a Long Short-Term Memory (LSTM) network was developed to detect intrusions in Internet of Vehicles (IoV) networks. The problem was formulated as a binary classification task, distinguishing between benign traffic and malicious attacks.

Before model training, a comprehensive data preprocessing pipeline was implemented:

* **Data Cleaning:** All missing values and infinite values were handled by replacing them with NaN and dropping affected rows to ensure clean and consistent data.
* **Feature Engineering:** To address skewness caused by extremely large values, a logarithmic transformation was applied selectively to positive feature values. Afterward, values were clipped within a reasonable range to further control outliers.
* **Feature Scaling:** Min-Max normalization was employed to scale the feature set between 0 and 1, improving the convergence of the LSTM model.
* **Label Encoding:** Labels ('Benign' and 'Attack') were encoded into binary numeric values (0 and 1).
* **Train-Test Splitting:** The dataset was split into training and testing subsets in an 80-20 ratio while maintaining class distribution through stratification.

The LSTM model was designed with the following architecture:

* **First LSTM Layer:** 128 units with return sequences enabled, capturing complex temporal relationships.
* **Dropout Layer:** 20% dropout to reduce overfitting by randomly deactivating neurons during training.
* **Second LSTM Layer:** 64 units with return sequences disabled, focusing on summarizing the sequential features.
* **Dense Layer:** A fully connected layer with 32 ReLU-activated neurons for additional abstraction.
* **Output Layer:** A single neuron with sigmoid activation to produce binary predictions.

The model was compiled with the Adam optimizer and binary cross-entropy loss function, appropriate for binary classification tasks. A ModelCheckpoint callback was used to save the best-performing model based on validation accuracy during training.

The network was trained over 20 epochs with a batch size of 64, utilizing the testing dataset as validation data.

**Model Evaluation**

After training, the model’s performance was evaluated on the test set. The evaluation process included:

* **Accuracy and Loss:** Overall test accuracy and loss were reported to gauge general performance.
* **Prediction Thresholding:** Model outputs, which are probabilities, were converted to binary labels by applying a threshold of 0.5.
* **Classification Report:** Detailed metrics such as precision, recall, and F1-score were provided for both benign and attack classes.
* **Confusion Matrix:** A confusion matrix was generated and visualized to examine the distribution of true positives, true negatives, false positives, and false negatives.

Additionally, training history plots were generated:

* **Accuracy Curve:** Illustrated the progression of training and validation accuracy over epochs, highlighting the model's learning stability.
* **Loss Curve:** Showed how the training and validation loss evolved, helping to detect possible overfitting or underfitting behaviors.

The model achieved strong performance with high accuracy and F1-scores, demonstrating its effectiveness in detecting intrusions in IoV environments. The confusion matrix confirmed that the majority of benign and attack instances were correctly classified.

All key evaluation artifacts, including the confusion matrix, classification report, and training curves, were saved for documentation and further analysis.

# **CHAPTER 5**

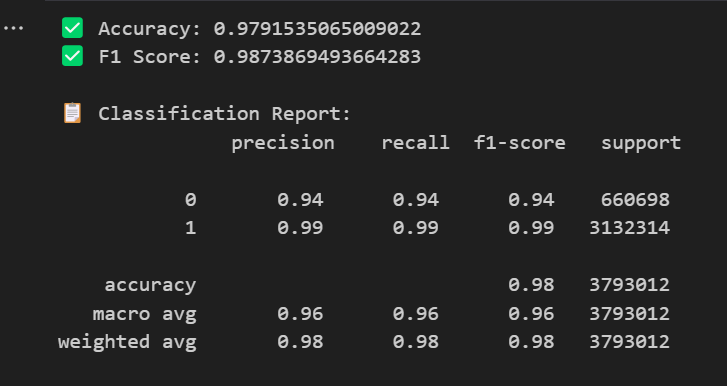
**RESULT AND DISCUSSION**

## **5. Result And Discussion**

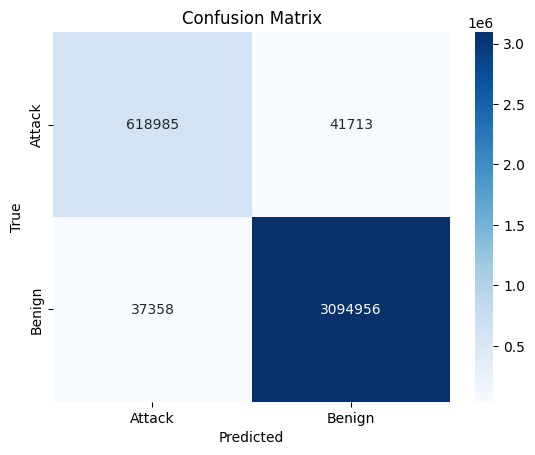
This chapter presents the performance results of the developed machine learning models for intrusion detection in the Internet of Vehicles (IoV) environment. Four different models were evaluated: Random Forest and LSTM models for both binary and multiclass classification. Their effectiveness was assessed using accuracy, F1-score, confusion matrix, and classification reports.

## **5.1 Random Forest: Binary Classification**

The Random Forest classifier achieved strong performance on the binary classification task, where the goal was to distinguish between benign traffic and malicious attacks.

* **Accuracy:** 97.91%
* **F1-Score:** 98
* **Confusion Matrix:** Most benign and attack samples were correctly classified, with very few misclassifications.

**Figure5.1 Output of Random Forest: Binary Classification**



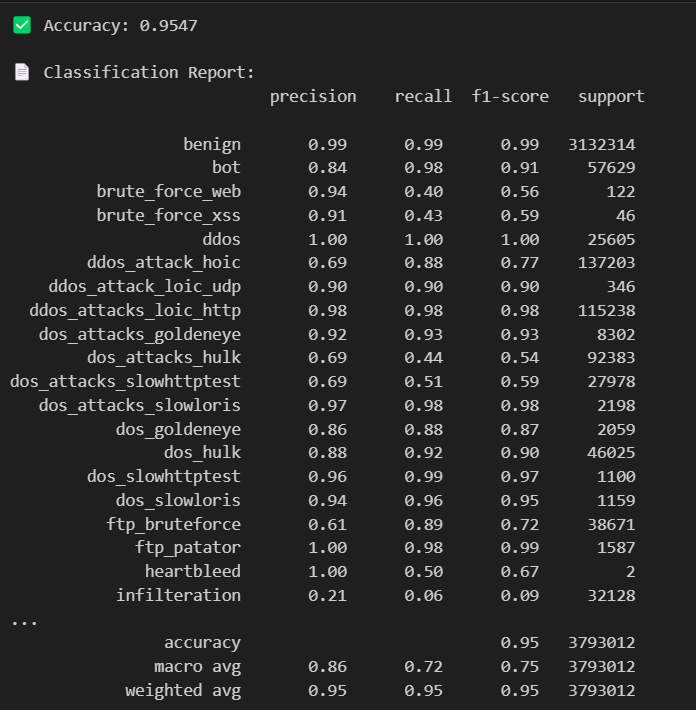
**Figure 5.2 Confusion Matrix of Random Forest: Binary Classification**

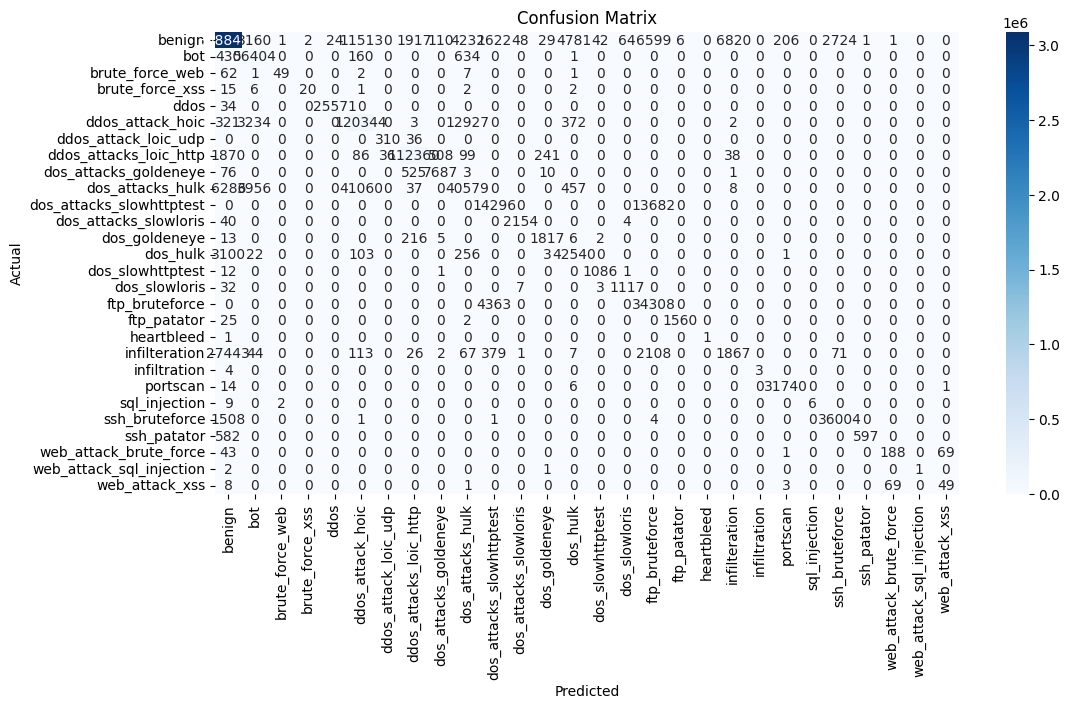
The high accuracy and low false positive rate suggest that Random Forest is highly effective for binary intrusion detection, offering both robustness and interpretability through feature importance analysis.

## **5.2 Random Forest: Multiclass Classification**

For the multiclass classification task, where each attack type needed to be individually identified, the Random Forest model also performed very well.

* **Accuracy:** 95.47%
* **Classification Report:** Showed strong precision and recall across most classes, though a few attack categories with similar patterns (e.g., Web Attack vs. Brute Force) showed slightly lower scores.
* **Confusion Matrix:** Correct classification was dominant, though slight misclassifications occurred among certain attack types.

**Figure 5.3 Output of Random Forest: Multiclass Classification** 



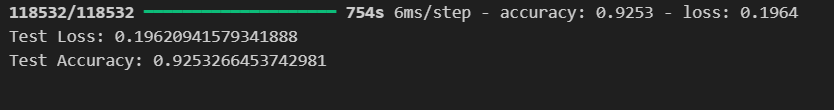
**Figure 5.4 Confusion Matrix of Random Forest: Multiclass Classification**

These results highlight Random Forest’s capability to handle high-dimensional multiclass data effectively, although subtle similarities between certain attack behaviors may lead to occasional confusion.

## **5.3 LSTM: Binary Classification**

The LSTM network, designed to capture sequential patterns, also demonstrated excellent results in binary classification.

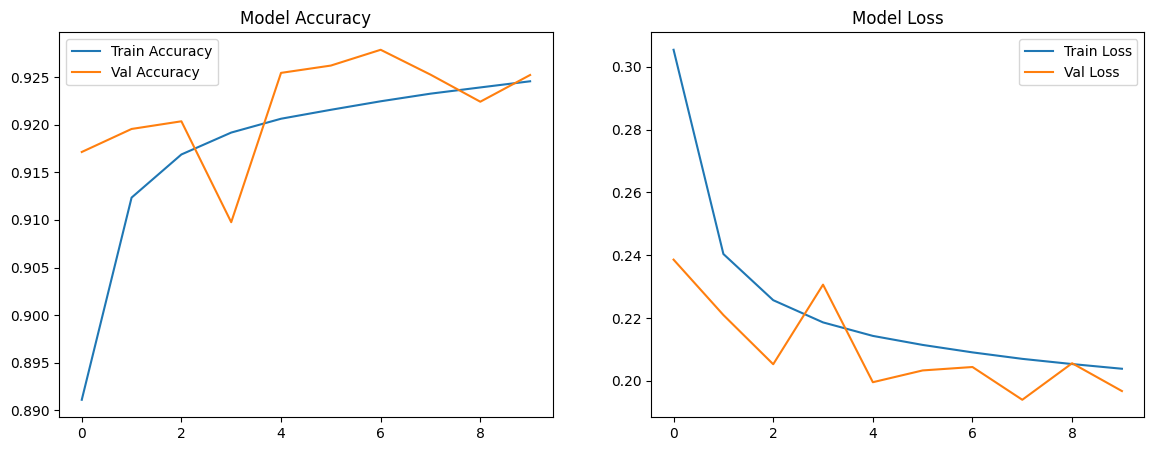
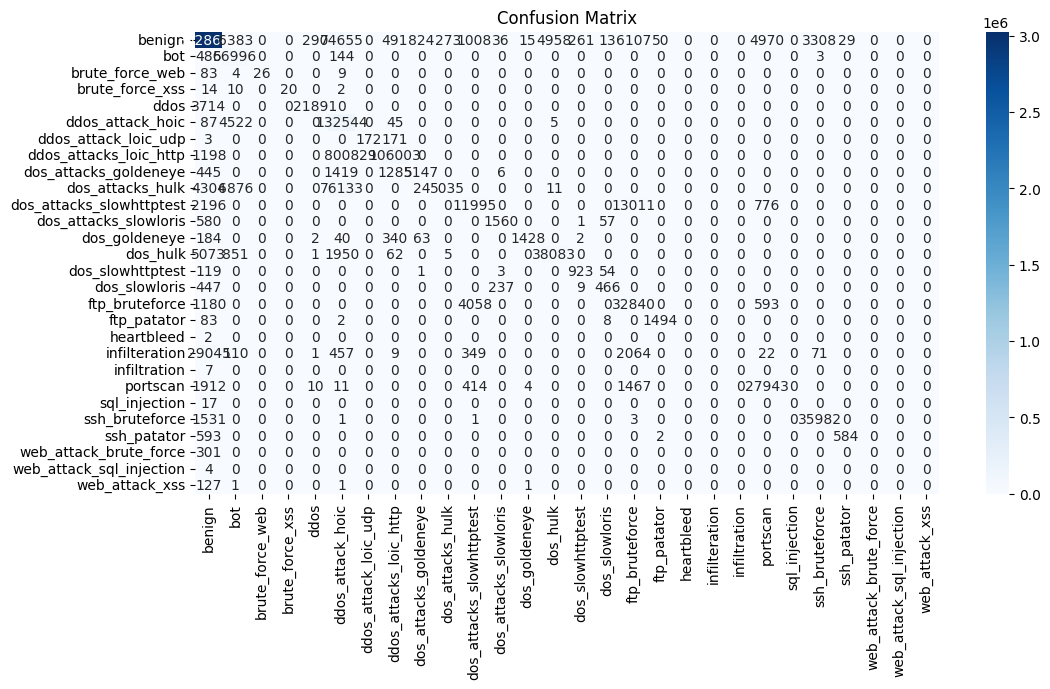
* **Accuracy:** 92%
* **F1-Score:** 89%
* **Training Curves:** The accuracy and loss curves confirmed that the model generalized well without significant overfitting.

**Figure 5.5 Output of LSTM: Binary Classification**

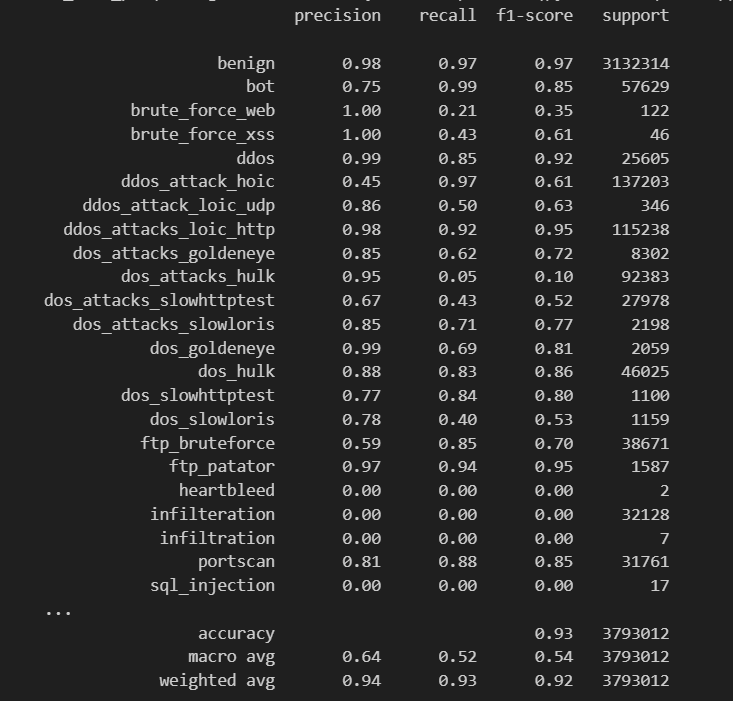
The LSTM model’s ability to capture temporal dependencies made it a strong candidate for real-time intrusion detection in dynamic IoV environments.

## **5.4 LSTM: Multiclass Classification**

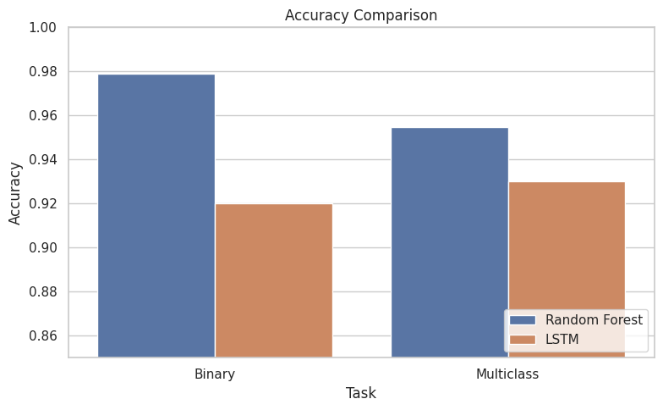
## In the multiclass setting, the LSTM model maintained competitive performance.

* **Accuracy:** 93%
* **Classification Report:** Precision and recall were strong for major attack classes but slightly lower for minority classes.
* **Confusion Matrix:** Most classes were correctly predicted; however, attacks with similar traffic behavior occasionally caused misclassifications.
* **Figure 5.6** **Loss in LSTM Multiclass classification**

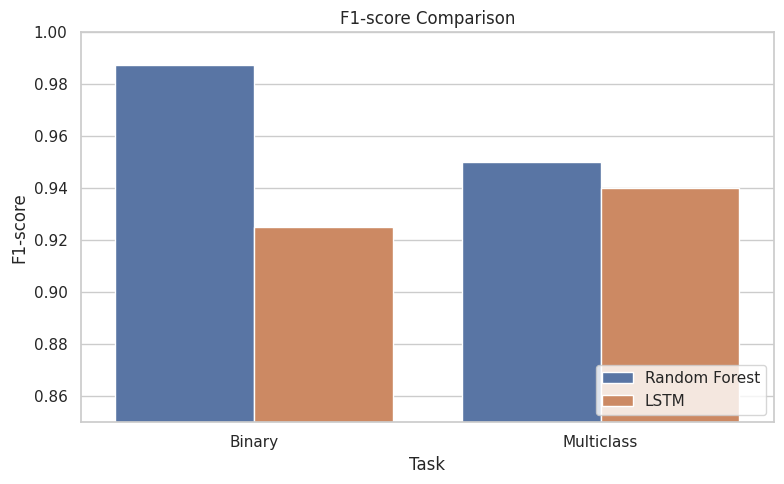
**Figure 5.7 Confusion Matrix of LSTM: Multiclass Classification**



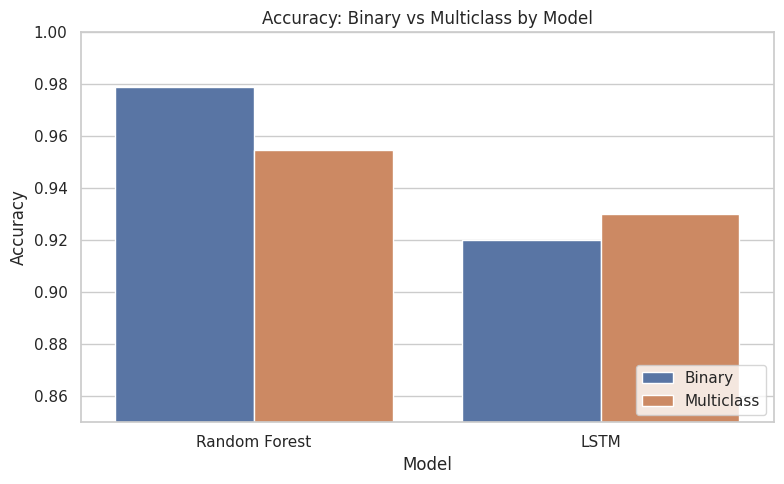
**Figure5.8 Output of LSTM: Multi Classification**

****Despite the increased complexity of multiclass prediction, the LSTM model was able to accurately identify most attack types, demonstrating the effectiveness of deep learning in modeling complex, sequential patterns in network traffic.

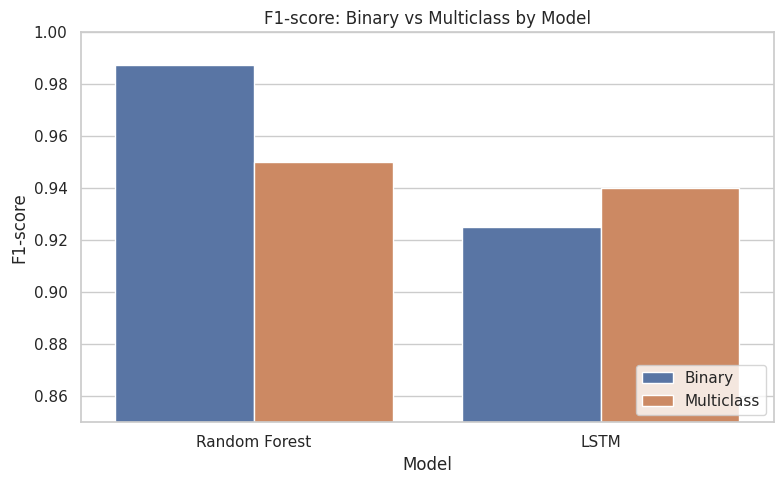
**Figure 5.9 Comparison of Accuracy in LSTM and RF**

****

**Figure 5.10 Comparison of F1-Score in LSTM and RF**

****

**Figure 5.11 Comparison of Accuracy of different mode in LSTM and RF**

****

**Figure 5.12 Comparison of F1-Score of different mode in LSTM and RF**

# 

# 

# **CHAPTER 6**

**CONCLUSION AND FUTURE WORK**

## **6.1 Conclusion**

The Internet of Vehicles (IoV) represents a significant advancement in smart transportation systems, offering improved safety, efficiency, and connectivity. However, the increased interconnectivity also introduces numerous security challenges, making intrusion detection systems (IDS) an essential component for safeguarding vehicular networks.

In this project, machine learning and deep learning techniques were employed to develop an IDS capable of detecting both binary (benign vs attack) and multiclass (different attack types) intrusions. Two distinct approaches were investigated:

* **Random Forest:** An ensemble-based machine learning algorithm known for its robustness, interpretability, and high accuracy.
* **Long Short-Term Memory (LSTM):** A deep learning model capable of learning complex sequential patterns in network traffic data.

Comprehensive experiments were conducted on a merged and preprocessed dataset combining CICIDS2017 and CICIDS2018 attack scenarios. The models were evaluated based on accuracy, F1-score, classification reports, and confusion matrices.

Both Random Forest and LSTM models demonstrated strong performance:

* Random Forest models achieved high accuracy for both binary and multiclass classification, with fast training times and high interpretability.
* LSTM models captured the sequential dependencies in network traffic effectively, achieving competitive results, especially in dynamic traffic scenarios.

These results confirm that both traditional machine learning and deep learning approaches have considerable potential for IoV intrusion detection applications. However, each method offers its own strengths and trade-offs in terms of speed, complexity, and predictive power.

## **6.2 Future Work**

Although the results achieved in this project are promising, several areas remain open for future exploration:

* **Hybrid Models:** Combining ensemble methods with deep learning architectures could leverage the strengths of both approaches, potentially improving detection accuracy and robustness.
* **Attention Mechanisms:** Integrating attention layers into LSTM networks could help the model focus on critical patterns, especially for multiclass intrusion scenarios.
* **Real-Time Detection:** Deploying the developed models on embedded vehicular systems to perform real-time intrusion detection would be an important step toward practical implementation.
* **Larger and Diverse Datasets:** Incorporating more recent datasets covering a wider range of attack types and network behaviors could improve model generalization to evolving threats.
* **Explainability and Interpretability:** Developing explainable AI (XAI) approaches for deep learning models can help in understanding the decision-making process, crucial for security-critical applications.

As the IoV landscape continues to evolve, future intrusion detection systems must adapt dynamically to new threats while maintaining low latency, high accuracy, and minimal resource consumption. Further research into lightweight, adaptive, and interpretable machine learning models will be key to advancing security in the next generation of intelligent vehicular networks.

# **REFERENCES**

1. S. Woo, H. Jo, and D. H. Lee, “TCAN-IDS: Temporal CAN Intrusion Detection System Using Convolutional Neural Networks,” *IEEE Access*, vol. 9, 2021.
2. P. Lee et al., “MTH-IDS: Multi-Tiered Hybrid Intrusion Detection System for Intra and Inter Vehicle Networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 2341–2350, 2021.
3. R. Sharma and S. Singh, “Deep Transfer Learning Techniques in Intrusion Detection System for Internet of Vehicles,” *Journal of Intelligent & Fuzzy Systems*, vol. 47, no. 2, pp. 1761–1774, 2024.
4. A. Khan, M. H. Rehmani, and M. Reisslein, “Machine Learning for Secure Vehicular Communication in the Internet of Vehicles: Recent Advances and Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1234–1260, 2021.
5. M. T. Hossain, M. Fotouhi, and R. Hassan, “Security of Autonomous Vehicles in 5G IoV: A Blockchain-based Approach,” *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1034–1045, 2022.
6. Y. Chen and J. Lin, “Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithms,” in *Proc. IEEE ICC*, 2024.
7. K. Liu et al., “Federated AI-Enabled In-Vehicle Network Intrusion Detection for IoV,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 12344–12356, 2022.
8. J. Kim, H. Park, and S. Lim, “FL-MAAE: Intrusion Detection for IoV with Federated Learning and Memory-Augmented Autoencoder,” *IEEE Access*, vol. 11, pp. 34567–34580, 2023.
9. Y. Wang et al., “Intrusion Detection in Intelligent Connected Vehicles Based on Weighted Self-Information,” *Sensors*, vol. 23, no. 3, pp. 1120, 2023.
10. A. Gupta and N. Kumar, “Modular Zero-Day Botnet Detection Using Isolation Forests and Particle Swarm Optimization,” *Computers & Security*, vol. 124, 2025.
11. M. Ahmed and B. Lee, “ATHENA: Physical Feature-Based Intrusion Detection System for Internet of Vehicles,” in *Proc. IEEE INFOCOM*, 2025.
12. S. Patel and A. Das, “Lifelong Learning IDS for 6G-IoV Using Class-Incremental and Federated Learning,” *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, 2024.
13. X. Zhu and Y. He, “A Multi-Stage Intrusion Detection System Based on Hierarchical Federated Learning in IoV,” *IEEE IoT Journal*, vol. 11, no. 2, pp. 3023–3035, 2024.
14. S. Banerjee and A. Roy, “Intrusion Detection in 5G-Enabled Internet of Vehicles Using Machine Learning,” *Ad Hoc Networks*, vol. 142, 2023.
15. A. Verma and M. Singh, “Hybrid IDS for IoT Devices Using LSTM-GRU Models,” in *Proc. IEEE ICMLA*, 2024.
16. R. K. Singh et al., “Multi-Stage IDS Framework for IoT-Enabled Internet of Vehicles,” *IEEE Access*, vol. 12, 2024.
17. Z. Fan and L. Yao, “Resilient Adaptive Event Trigger-Based Intrusion Detection in IoV,” *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, 2024.
18. T. Nguyen and J. Zhang, “AI-Based Intrusion Detection Systems: A Survey,” *IEEE Access*, vol. 12, pp. 34989–35010, 2024.
19. L. Wang and X. Huang, “Dynamic Hierarchical IDS on Edge Platform for IoV,” *Journal of Systems Architecture*, vol. 140, 2024.
20. M. Islam et al., “Enhancing Intrusion Detection in IoV through Federated Learning and Data Balancing Techniques,” *IEEE Transactions on Industrial Informatics*, vol. 19, no. 9, pp. 11200–11210, 2023.
21. S. Raj and V. K. Sharma, “A Survey on Federated Learning for Intrusion Detection in IoV,” *Computer Networks*, vol. 225, 2023.Z
22. Y. Luo and B. Zhang, “Interpretable Intrusion Detection in IoV Using Discrete Optimization Learning,” in *Proc. IEEE TrustCom*, 2024.
23. P. Yu and H. Zhou, “Optimized Fick’s Law Graph Point Intrusion Detection for Industrial IoV,” *Future Generation Computer Systems*, vol. 147, 2025.
24. M. Li, K. Xue, and D. S. Wei, “StatGraph: Multi-view Statistical Graph Learning for In-Vehicle IDS,” *IEEE Transactions on Dependable and Secure Computing*, 2023.
25. J. Rana and S. A. Hussain, “Enhancing IDS Performance on Edge of Things,” *Computer Communications*, vol. 204, 2023.
26. K. Saxena and V. Saini, “Innovative IDS for High-Density Communication Networks Using AI,” *International Journal of Information Security*, vol. 22, no. 1, 2023.
27. M. Hossain et al., “Zero-X: Blockchain-Enabled Open-Set Federated Learning Framework for Zero-Day Attack Detection in IoV,” *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, 2024.
28. R. Zhou and Y. Deng, “Homomorphic Encryption-Enabled Federated Learning for Privacy-Preserving IDS in Resource-Constrained IoV Networks,” *Computer Networks*, vol. 245, 2024.
29. T. J. Wong and P. H. Yu, “Deep Learning-Enabled IDS for Industrial IoT,” *Sensors*, vol. 24, no. 1, 2024.
30. H. Liu et al., “Dynamic Hierarchical IDS for IoV on Edge Computing Platform,” *Journal of Parallel and Distributed Computing*, vol. 180, 2024.
31. A. R. Pandey and R. Agrawal, “Evolutionary LightGBM-Based IDS for IoT Networks,” *Applied Intelligence*, vol. 65, no. 2, 2025.
32. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116. [Online]. Available: <https://www.unb.ca/cic/datasets/ids-2017.html>
33. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "CICIDS2018: Toward a realistic benchmark for network intrusion detection systems," *Canadian Institute for Cybersecurity*, University of New Brunswick, 2018. [Online]. Available: <https://www.kaggle.com/datasets/sampadab17/cicids2018>
34. M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm and Random Forest Classifier," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, Oct. 2016.
35. S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," *Technical Report No. 99-15, Department of Computer Engineering, Chalmers University of Technology*, March 2000.