Artificial Intelligence Enabled Vehicular Vision and Service provisioning for Advanced Driver Assistance Systems (ADAS)

THESIS

Submitted in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

by

Amit Umesh Chougule

ID. No. 2020PHXF0415P

Under the Supervision of

Prof. Vinay Chamola

Associate Professor, Department of Electrical and Electronics Engineering

And

Under the Co-Supervision of

Prof. Pratik Narang

Associate Professor, Department of Computer Science & Information Systems

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

2024

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis titled "**Artificial Intelligence Enabled Vehicular Vision and Service provisioning for Advanced Driver Assistance Systems (ADAS)"** submitted by **Mr. Amit Umesh Chougule** ID No **2020PHXF0415P** for award of Ph.D. of the Institute embodies original work done by him under our supervision.

_____

Signature of the Supervisor:
Name: **Prof. Vinay Chamola**
Designation: **Associate Professor**

_____

Signature of the Co-Supervisor:
Name: **Prof. Pratik Narang**
Designation: **Associate Professor**

Date: 20 April, 2024

Acknowledgments

My academic journey towards completing this doctoral study has been enriched by the unwavering support and guidance of numerous individuals and institutions, and for this, I am profoundly grateful. At the forefront of this acknowledgment, I extend my deepest appreciation to my distinguished thesis advisor, Prof. Vinay Chamola. His relentless commitment to excellence, continuous guidance, and meticulous supervision have played an instrumental role in shaping the methodological framework and the execution of this research throughout my doctoral studies. Prof. Chamola embodies the spirit of an advisor who not only provided invaluable insights but also empowered me with the freedom and agility to explore and conduct my research with rigor and innovation.

I am equally grateful to Prof. Pratik Narang, who co-supervised my thesis, for his invaluable guidance and visionary contributions regarding the perception of vehicles. His mentorship has significantly enhanced the depth and breadth of my research. Furthermore, I must acknowledge the members of my Dissertation Advisory Committee (DAC), Prof. K. K. Gupta and Dr. G. S. S. Chalapathi, whose timely feedback and insightful suggestions have been integral from the proposal stage and throughout the progress of my research. The interactive research seminars with them have provided me with invaluable opportunities to standardize and elevate the quality of my work.

In addition, I express my profound thanks to Prof. F. Richard Yu, a Fellow of IEEE at Carleton University, Ottawa, Canada, for his guidance on autonomous vehicle technology and for providing access to resources from the TrustCAV Lab during my visit as a visiting researcher to Carleton University. I would also like to acknowledge the valuable guidance and time dedicated by Prof. Amir Hussain, a Senior Member of IEEE at Edinburgh Napier University, in support of my research on cognitive and brain-related studies.

I take this moment to express my gratitude to Prof. Navneet Gupta and Prof. H. D. Mathur, as well as Prof. Praveen Kumar A.V, for their support and for granting me the freedom to engage in research within the EEE department during their respective tenures. I express my gratitude to Prof. V. Ramgopal Rao (Vice-Chancellor) and Prof. Sudhirkumar Barai (Director) for providing me the opportunity to pursue my PhD degree at BITS Pilani. Moreover, my thanks extend to the Shastri Indo-Canadian Institute (SICI), supported by funds from the Ministry of Education (MoE), Government of India, for providing the fellowship that significantly bolstered my research studies.

In the collective pursuit of knowledge and innovation, these individuals and institutions have been my pillars of support, and I am deeply appreciative of their contributions to the success of this doctoral thesis.

Amit Chougule

Abstract

The recent advancements in the automotive and transportation industry have ushered in a new era of travel, marked by increased safety and comfort for passengers. These improvements have been made possible through the integration of technology and sensors into vehicles. This evolution from traditional automobiles to the current generation of autonomous vehicles signifies the tremendous progress in vehicle technology. In the past, human drivers were responsible for all tasks related to driving, but today, we have semi-autonomous, less driver-dependent vehicles that operate with minimum human intervention.

However, this integration of technology and sensors, while enhancing the capabilities of vehicles, has introduced a new set of challenges. Modern vehicles are equipped with a multitude of sensors and communication protocols, making them susceptible to cyberattacks and security threats. Through these interconnected channels, malicious actors can potentially gain control of a vehicle, leading to malfunctions and, in the worst cases, accidents, injuries, and even fatalities. Such attacks can disrupt the Vehicle Ad-Hoc Network (VANET) environment, posing significant risks to passengers, drivers, pedestrians, and other vehicles sharing the road.

In the contemporary automotive landscape, the deployment of Advanced Driver Assistance Systems (ADAS) stands as a pivotal advancement, reshaping vehicular safety and functionality. Remarkably, within the expansive realm of one billion global vehicles, a mere 10% are adorned with the sophisticated ADAS technology. This statistic underscores the pervasive integration of ADAS, with approximately 100 million vehicles currently navigating the world's thoroughfares equipped with this cutting-edge technology. The ascendancy of ADAS to such a significant proportion within the vehicular ecosystem heralds a transformative era in road safety and underscores its substantive presence on a global scale. ADAS-enabled ve-

hicles are more prevalent on the road compared to fully autonomous vehicles. ADAS, typically representing Level 2 automation, provides valuable driver assistance and partial automation. In Level 2, vehicles can assume control of steering, acceleration, and braking in specific situations, but it's essential to emphasize that driver supervision remains a critical requirement. Addressing the challenges associated with ADAS technology and enhancing its capabilities is a fundamental step before progressing to higher levels of vehicle automation, particularly considering the widespread presence of ADAS-enabled vehicles on today's roadways.

This thesis proposes Artificial Intelligence Enabled Vehicular Vision and Service Provisioning for Advanced Driver Assistance Systems (ADAS), which takes on the multifaceted issues that exist within ADAS-enabled vehicles, including security vulnerabilities, improving perception in adverse weather conditions, and addressing driver inattention problems. The solutions and insights presented in this thesis are designed to tackle these challenges and fortify ADAS technology, ultimately working towards a future that is safer, more efficient, and better equipped to meet the demands of modern transportation.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Term	Definition
ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AV	Autonomous Vehicle
CAN	Controller Area Network
CNN	Convolutional Neural Networks
DOS	Denial Of Service
ECU	Electronic Control Units
EEG	Electroencephalogram
GAN	Generative Adversarial Network
GHG	Greenhouse gases
IDS	Intrusion Detection System
IoV	Internet of Vehicles
ITS	Intelligent Transportation System
LIN	Local Interconnect Network
LSTM	Long Short-Term Memory
MSE	Mean Square Error
OBU	On-Board Unit
ROC	Receiver Operating Characteristic
RSU	Road-Side Unit
RNN	Recurrent Neural Networks
VANET	Vehicular Adhoc Network
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrian
V2S	Vehicle to Sensor
V2V	Vehicle to Vehicle
V2X	Vehicle-to-everything

Chapter 1

Introduction

In this chapter, the foundation is established by delving into the background and emphasizing the importance of Advanced Driver Assistance Systems (ADAS) technology within the Vehicle Ad-Hoc Network (VANET) environment. The central focus is on elucidating the various challenges inherent in ADAS-enabled vehicles, ranging from security vulnerabilities to addressing driver inattention, understanding the behavior of ADAS vehicles in traffic at intersections, and enhancing perception in adverse weather conditions.

1.1 Advanced Driver Assistance Systems-Enabled Vehicles

In the contemporary era of Intelligent Transportation Systems (ITS), the remarkable advancements within the automobile industry have ushered in an era of safer and more comfortable journeys for humanity. The trajectory of the automobile industry has evolved from conventional, manually operated vehicles to the cutting-edge autonomous cars of today. These autonomous vehicles (AV) exhibit a spectrum of automation levels (Figure 1.1), ranging from no automation at Level Zero

to full automation at Level Five, where self-driving capabilities are realized. In the expansive realm of vehicular technology, a salient observation emerges regarding the prevalence of ADAS. Within the vast global automotive fleet, a mere 10% of vehicles are endowed with ADAS technology, as elucidated by recent statistics [11, 12]. This statistic underscores the presence of approximately 100 million vehicles on the road that are fortified with ADAS capabilities. The significance of this numerical magnitude cannot be overstated, as it encapsulates a substantial portion of the automotive landscape, illuminating the noteworthy integration and adoption of ADAS within contemporary vehicular systems.

ADAS technology represents Level 2 of automation, offering valuable driver assistance and partial automation. In Level 2, vehicles can assume control over steering, acceleration, and braking in specific scenarios [1]. However, it is crucial to emphasize that the driver's active supervision remains a requisite at all times. Given the prevalent prevalence of ADAS-enabled vehicles in the modern landscape, surpassing the challenges inherent in ADAS technology and reinforcing its capabilities stand as imperative tasks prior to advancing to the next echelons of automation. These endeavors are instrumental in ensuring the seamless and secure progression toward higher levels of vehicle automation.

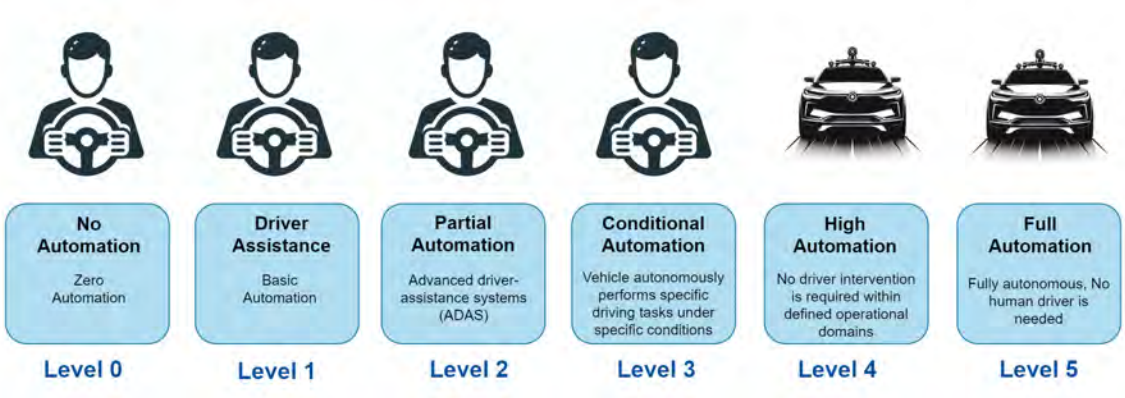


Figure 1.1: Levels of Autonomy [1]

1.1.1 Security Provisioning for ADAS-Equipped Vehicles in VANET Environments

In the current landscape, ADAS-enabled vehicles face a range of vulnerabilities, leaving them susceptible to various attackers and hackers [13–15]. These malicious actors have the capacity to compromise the security of ADAS-equipped cars, potentially enabling remote control or inducing malfunctions in vehicle operations. Such breaches pose significant threats to the safety and well-being of passengers and drivers. Furthermore, if ADAS-enabled vehicles were to fall victim to these attacks, the potential repercussions could extend beyond individual cars, possibly leading to chaotic situations within the VANET environment. These situations, in turn, could imperil the safety of surrounding vehicles and pedestrians.

Among the various vehicular attacks, the Masquerade injection attack [13] is notably straightforward to implement on Advanced Driver Assistance Systems (ADAS). This form of attack involves deceiving an ADAS by presenting falsified information with the intent to distort the vehicle's perception or decision-making capabilities. The primary targets of Masquerade injection attacks are the sensors and software critical to ADAS operations, including cameras, radar, LiDAR, and the systems responsible for data processing and communications within the vehicle. This attack encompasses methods such as Sensor Spoofing, Data Manipulation, GPS Spoofing, and exploiting Software Vulnerabilities. The implications of such attacks are severe, leading to operational disruptions and significant safety risks, thereby compromising the integrity and reliability of the ADAS-enabled vehicle. Similarly, the ADAS system is susceptible to other forms of attacks, such as attacks on the CAN bus, wheel steering attacks [15], acceleration attacks [15], and context-aware attacks [16]. Therefore, safeguarding ADAS-enabled vehicles from a spectrum of attacks through effective threat detection mechanisms is of paramount importance. To fortify the security of ADAS-enabled vehicles, a mul-

tifaceted approach is essential. This approach necessitates securing in-vehicle operations while concurrently safeguarding the VANET environment in which these vehicles operate. To this end, it is imperative to bolster in-vehicle communication to ensure the safety of the vehicle itself. Additionally, the implementation of edge-level intrusion detection systems assumes critical significance [17]. Such systems can be deployed at various points, including On-board units (OBUs), road-side units (RSUs), or other strategically positioned nodes within the VANET. By doing so, the VANET environment, along with the vehicles within it, can be effectively safeguarded, mitigating the risks associated with cyber threats and enhancing overall security within the vehicular ecosystem.

1.1.2 Effect of Microsleep on Driver's Attention in ADAS

In the contemporary landscape of modern vehicles, a hierarchical classification system based on the level of autonomy has emerged. Traditional vehicles fall within Level 0, whereas self-driving cars, achieving full automation and driverless operation, are designated as Level 5 [1]. However, the majority of vehicles on the road today are ADAS-enabled, placing them at Level 2 of autonomy, characterized by partial automation [1]. In Level 2 vehicles, the presence of a driver behind the steering wheel is imperative, unlike fully autonomous cars, which operate without human intervention.

The ADAS-based vehicles offer a range of functionalities, including Adaptive Cruise Control (ACC), lane-keeping assistance, and more. Nevertheless, there are instances where the car's lane-keeping or ACC functionality may falter or require human intervention. In such cases, the responsibility shifts to the driver, emphasizing the importance of the driver's role in ensuring the proper operation of ADAS-equipped vehicles [18–20]. Despite the advanced assistance provided by ADAS-enabled vehicles, a significant number of accidents are still attributable to

various human factors. Driver-related issues, such as inattentiveness, drowsiness, fatigue, or insufficient sleep, account for a substantial portion of road incidents. Among these, microsleep episodes emerge as a particularly complex and significant contributing factor to accidents, presenting a formidable challenge in terms of detection and mitigation [21–23]. These episodes involve brief, involuntary lapses in attention, lasting only a few seconds, yet they can have catastrophic consequences. Microsleeps render the driver temporarily unaware of their surroundings, incapacitating them from reacting to sudden or unexpected events. This issue underscores the pressing need for an effective solution to detect, manage, and ultimately prevent microsleep episodes among drivers, thereby enhancing the safety of both the driver and other road users.

1.1.3 ADAS-Enabled Vehicles at Intersections with Traffic Congestion

Intersections are pivotal junctures in urban traffic systems where vehicles and pedestrians converge, often leading to traffic congestion when the volume surpasses the intersection's capacity [24–26]. Ineffective traffic management at intersections results in standstills and sluggish movement, potentially causing gridlock that extends beyond the intersection itself. Uninterrupted traffic flow through intersections is crucial for efficient and safe transportation. Addressing congestion here is imperative for enhancing urban transportation systems. It arises from a complex interplay of factors, including the exceeding of intersection capacity, limitations of fixed-time traffic signals, and driver behaviors like reckless driving. Additionally, idling at intersections can affect drivers' physiological well-being, impacting alertness and reaction times [27]. Furthermore, relying on outdated traffic data for planning rather than real-time information can lead to suboptimal traffic management. Traditional traffic signals are increasingly struggling to

handle modern urban traffic, operating on predetermined schedules and lacking real-time adaptability [28]. In contrast, Advanced Driver Assistance System (ADAS)-enabled vehicles with advanced sensors, real-time data sharing, and communication with intelligent traffic signals are poised to revolutionize intersection dynamics and alleviate congestion, offering safer and more efficient urban mobility [29–31]. As the prevalence of ADAS systems grows, their impact on intersection dynamics and traffic congestion is set to reshape urban transportation. To bridge the technology gap, the deployment of intelligent traffic signals within VANETs offers a promising solution that benefits both ADAS-enabled and traditional vehicles, optimizing signal timings, reducing wait times, and enhancing overall traffic flow for a more efficient and sustainable urban mobility landscape [11].

1.1.4 Perception for ADAS-Enabled Vehicles

The operational efficacy of ADAS-enabled vehicles is paramount in ensuring safe and comfortable journeys for both passengers and drivers. ADAS serves as a multifaceted tool, providing various features and acting as a reliable driver assistant. Under normal clear weather conditions, ADAS functions seamlessly; however, adverse weather conditions pose a significant challenge to its operability [32]. ADAS-equipped vehicles are equipped with cameras that analyze the surrounding environment, offering valuable information to the driver during the journey. These cameras play a pivotal role in enhancing driver awareness and vehicle safety. They enable features such as object detection, lane departure warning, traffic sign recognition, forward and backward collision warnings, adaptive cruise control, parking assistance, driver monitoring, as well as pedestrian and obstacle detection. However, adverse weather, especially fog, significantly hampers the camera's functionality [33–36]. Fog obscures the view, rendering the front areas unclear, blurry, and concealing nearby objects [37]. In such foggy conditions,

the vehicle's ability to assist the driver becomes severely compromised, impeding its effectiveness in providing crucial support during driving. To overcome this challenge, the development of mechanisms capable of clearing fog or haze in any surrounding or image is imperative. These mechanisms are essential for generating dehazed images, enabling subsequent computer vision and ADAS operations to function effectively in hazy environments. By providing clear visibility in adverse weather conditions, these dehazing techniques are integral to ensuring the continuity of ADAS functionality, enhancing driver safety, and augmenting the overall reliability of intelligent transportation systems.

1.2 Motivation and Objectives

The integration of technology into the automobile industry has indeed revolutionized traditional driving and vehicles, enhancing safety, comfort, and efficiency. However, this progress is not without its challenges and limitations that continue to persist in the automobile industry. Remarkably, within the expansive realm of one billion global vehicles, a mere 10% are adorned with sophisticated ADAS technology [11]. This statistic underscores the pervasive integration of ADAS, with approximately 100 million vehicles currently navigating the world's thoroughfares equipped with this cutting-edge technology. The ascendancy of ADAS to such a significant proportion within the vehicular ecosystem heralds a transformative era in road safety and underscores its substantive presence on a global scale. This underscores the imperative of addressing the current limitations of ADAS-equipped vehicles before transitioning completely from ADAS technology (level 2) to the next level of autonomy, which stands at level 3 [1]. The journey to the next level of autonomy necessitates not only overcoming the present limitations of ADAS technology but also enhancing its efficiency. This entails a comprehensive effort to bolster the capabilities and reliability of ADAS systems,

ensuring their readiness to navigate the complexities of urban transportation and paving the way for the safe and seamless transition to more advanced levels of automation. The following are the objectives of the thesis,

1. To enhance the security of ADAS-equipped vehicles in VANET environments through the creation of a state-of-the-art deep learning-based Intrusion Detection System (IDS) framework.
2. Improving driver's on-road attentiveness through the detection of short-term Microsleep using EEG and deep learning techniques in ADAS.
3. Designing a Novel Framework for Traffic Congestion Management at Intersections ADAS vehicles in in VANET environment.
4. Design a cutting-edge computer vision model for robust image dehazing under adverse weather conditions.

1.3 Organization of the Thesis

This thesis is dedicated to the advancement of Artificial Intelligence-driven Vehicular Vision and Service Provisioning, with a specific focus on strengthening ADAS. The introductory Chapter 1 provides a comprehensive overview of ADAS-enabled vehicles, delving into the motivation behind this research and elucidating the research objectives. Chapter 2 offers an extensive exploration of the literature survey, encompassing various facets of ADAS-enabled vehicles and VANET. It lays the groundwork for understanding the state of the art in this domain. Chapter 3 delves into the critical area of Security Provisioning in VANETs for ADAS-Equipped Vehicles. It provides detailed insights into the developed models, including MbRE for VANET edge scenarios, the HybridSecNet designed for in-vehicle communication, specifically the CAN protocol, and the SCAN-GAN model, which leverages generative adversarial networks to synthesize datasets

crucial for training deep learning-based security models. This chapter also offers a glimpse into hardware resource utilization on different devices and showcases the rapid intrusion detection capabilities of the MbRE model. Chapter 4 uncovers an EEG-based deep learning approach designed to detect short-term episodes of driver microsleep in ADAS-enabled vehicles, emphasizing its potential to enhance on-road driver attention. The chapter also provides insights into the hardware implementation of the proposed pipeline. Chapter 5 delves into the future prospects of ADAS-Enabled Vehicles at Intersections, with a particular focus on addressing traffic congestion challenges. Chapter 6 introduces a computer vision-based mechanism aimed at dehazing images and enhancing image clarity. The proposed AGD-NET model takes center stage, and the chapter elaborates on its functioning. Furthermore, it presents comparative results through quantitative and qualitative analyses, demonstrating the superior dehazing capabilities of the AGD-NET model when compared to other state-of-the-art dehazing models. Finally, Chapter 7 serves as a culmination of the work presented in this thesis, offering a summary of the research carried out and deriving conclusions based on the findings and contributions presented in the preceding chapters.

1.4 Bibliographic Note

Parts of Chapter 3 appear in the following journal papers,

- Chougule, Amit, Varun Kohli, Vinay Chamola, and Fei Richard Yu. "Multi-branch reconstruction error (mbre) intrusion detection architecture for intelligent edge-based policing in vehicular ad-hoc networks", *IEEE Transactions on Intelligent Transportation Systems*, 2022.
  DOI: https://doi.org/10.1109/TITS.2022.3201548

- Chougule, Amit, Ishan Kulkarni, Tejasvi Alladi, Vinay Chamola, and Fei Richard Yu. "HybridSecNet: In-Vehicle Security on Controller Area Net-

works through a Hybrid Two-Step LSTM-CNN Model", *IEEE Transactions on Vehicular Technology*

- Chougule, Amit, Kartik Agrawal, and Vinay Chamola. "SCAN-GAN: Generative Adversarial Network Based Synthetic Data Generation Technique for Controller Area Network", *IEEE Internet of Things Magazine* 6, no. 3 (2023): 126-130.
  DOI: https://doi.org/10.1109/IOTM.001.2300013

Parts of Chapter 4 appear in the following journal papers,

- Chougule, Amit, Jash Shah, Vinay Chamola, and Salil Kanhere. "Enabling Safe ITS: EEG-Based Microsleep Detection in VANETs", *IEEE Transactions on Intelligent Transportation Systems*, 2022.
  DOI: https://doi.org/10.1109/TITS.2022.3230259

Parts of Chapter 5 appear in the following journal papers,

- Chougule, Amit, Vinay Chamola, Vikas Hassija, Pranav Gupta, and F. Richard Yu. "A Novel Framework for Traffic Congestion Management at Intersections Using Federated Learning and Vertical partitioning", *IEEE Transactions on Consumer Electronics*, 2023.
  DOI: https://doi.org/10.1109/TCE.2023.3320362

Parts of Chapter 6 appear in the following journal papers,

- Chougule, Amit, Agneya Bhardwaj, Vinay Chamola, and Pratik Narang. "Agd-net: Attention-guided dense inception u-net for single-image dehazing." *Cognitive Computation* 16, no. 2 (2024): 788-801.
  DOI: https://doi.org/10.1007/s12559-023-10244-2

Chapter 2

Literature Survey

This chapter provides an extensive review of a multitude of Vehicular Vision and security Service provisioning schemes developed for VANETs. It offers a comprehensive exploration of existing approaches, highlighting their contributions and limitations in the realm of enhancing vehicular vision and security services within the VANET ecosystem. The review is instrumental in providing a comprehensive understanding of the state-of-the-art in this domain, paving the way for the development of novel and effective solutions.

2.1 Security Provisioning for Vehicles in VANET Environments

2.1.1 Edge-Assisted Vehicular Network Security

The extensive vulnerabilities inherent in VANETs [38–40] render these networks susceptible to a multitude of cyber threats [41, 42]. Malicious vehicular nodes may engage in misbehavior by disseminating inaccurate position or velocity data to other nodes within the network. Furthermore, the network may be inundated

with messages, leading to a denial of legitimate communication. Intrusive nodes may also compromise the privacy and integrity of data by collecting information from other nodes and broadcasting it as their own. Such intrusions within the network pose significant risks with potentially fatal consequences. Consequently, numerous studies have been conducted in the past decade to address intrusion detection [43], employing a diverse array of tools ranging from statistical techniques [44], blockchain solutions [45], machine learning approaches [46], to deep learning methods [47]. These multifaceted strategies underscore the ongoing efforts to fortify the security of VANETs against an evolving landscape of cyber threats.

The stringent security requirements of vehicular networks necessitate a departure from cloud platforms due to their high latency. Deep learning, renowned for its proficiency in handling substantial data volumes within VANETs [48, 49], aligns harmoniously with the principles of edge and fog computing, ensuring optimal resource utilization [50, 51]. As an illustration, a Convolutional Neural Network (CNN)-based Intrusion Detection System (IDS) framework exhibits remarkable precision in processing vehicular data and identifying a spectrum of anomalies [52]. Another innovative approach leverages blockchain and edge-based learning to classify diverse anomalies [45]. A multi-tiered hybrid IDS system is adept at detecting multiple attacks [53]. Furthermore, a Generative Adversarial Network (GAN)-based IDS designed for fog environments offers sustained long-term security [54]. Notably, a reconstruction and thresholding-based IDS system, employing Long Short-Term Memory (LSTM) techniques, is engineered to target specific attacks [55]. The work by Adhikari *et al.* [56] meticulously delineates a spectrum of security challenges inherent in an edge-centric intelligent Internet of Vehicles (IoV) framework. This framework, characterized by distributed smart vehicles and remote processing units, confronts a myriad of security attack modes, as elucidated in their comprehensive analysis. Furthermore, the study conducted by Onieva *et al.* [17] intricately investigates the potential leverage of enticing features

introduced by edge computing paradigms for enhancing security and privacy in the critical domain of vehicular networks (VNs). Particularly, the research sheds light on the adept exploitation of edge computing to address extant authentication and revocation challenges within the intricate fabric of VNs. These approaches collectively signify the invaluable role of deep learning and edge computing in fortifying the security of vehicular networks.

2.1.2 In-Vehicle Communication Security

In-vehicle communication, which entails communication among various subsystems within a vehicle, plays a pivotal role in modern automotive systems. The security of in-vehicle communication is paramount to ensure the integrity and confidentiality of data exchanged among these subsystems. A range of communication protocols is employed for in-vehicle communication, including LIN (Local Interconnect Network), FlexRay, Ethernet, and CAN (Control Area Network). Rajnák *et al.* [57] conducted a comprehensive analysis of the LIN protocol in vehicles, exploring various development tools and software interfaces. Their work contributes to understanding the intricacies of LIN-based in-vehicle communication. Milbredt *et al.* [58] proposed a Switched FlexRay mechanism designed to enhance effective bandwidth and safety for FlexRay networks, addressing key performance aspects in this communication protocol. The security of Ethernet in automotive settings is addressed in the work by Kim *et al.* [59], where they emphasize the importance of securing Ethernet communication within vehicles. Carnevale *et al.* [60] proposed a Macsec-based security solution for automotive Ethernet, providing a robust method to safeguard data integrity and confidentiality in Ethernet-based in-vehicle communication. Security mechanisms for the widely used CAN protocol have also been explored. An innovative method for detecting in-vehicle network intrusions is presented by [61], utilizing the analysis of time intervals between CAN signals to identify potential malicious activities. Tariq *et al.* intro-

duced CANtransfer [62], a novel approach leveraging transfer learning to enhance the detection performance for novel attacks by transferring knowledge from one set of attacks to another. Rathore *et al.* [63] conducted a comprehensive survey on cybersecurity for in-vehicle communication, addressing existing challenges and presenting solutions proposed by various researchers. Similarly, Martínez-Cruz *et al.* [64] provided a review of security considerations for in-vehicle communication protocols, discussing issues, challenges, and potential solutions. These studies collectively contribute to the evolving landscape of securing in-vehicle communication, ensuring the robustness of automotive systems in the face of evolving cyber threats.

2.1.3 Synthetic Data for Model Training for VANET Security

In the domain of VANET security using artificial intelligence, researchers have developed various technologies such as deep learning, generative adversarial networks, diffusion models, and more. Alladi, Tejasvi, *et al.* [47] introduced a deep learning-based classification framework for securing the Internet of Vehicles. Kaur, Gurjot, and Deepti Kakkar [65] presented a trust-based secure routing with a deep learning-based attack detection mechanism for VANET. Belenko, Viacheslav, *et al.* [66] proposed a GAN-based approach for intrusion detection. Similarly, Sreerag, V. *et al.* [67] put forth a GAN-based approach to detect various attacks. Some researchers have also explored diffusion-based security approaches, such as the work by Tang, Bin, *et al.* [68] which introduced a diffusion model-based network intrusion detection method. Wang, Yingshu, *et al.* [69] suggested an intrusion detection method based on denoising diffusion probabilistic models. However, the development of these models necessitates a substantial amount of data for training and evaluation. In the field of VANET security, there are only a limited number of datasets available, including the Veremi dataset [70], Car hacking dataset [71], V2X Security Threats datasets [72], and a few more. Nevertheless,

these datasets often lack the necessary volume, and newer techniques like generative adversarial networks require extensive data for effective model training. Insufficient data volume can lead to inadequate model training and present challenges for model evaluation since the limited data must be divided for training, validation, and testing. As a result, the scarcity of datasets and limited data volume becomes a major hurdle in developing new security mechanisms. To address this issue, many researchers have proposed data augmentation approaches during model development [73–75]. However, data augmentation techniques, while helpful, may not completely resolve the problem of limited data [76, 77]. This limitation underscores the ongoing challenges in dealing with data scarcity in the development of VANET security solutions.

2.2 Effect of Microsleep on Driver's Attention

The field of sleep analysis has seen remarkable progress in recent years, with a focus on understanding sleep-related activities and transitioning patterns to discern sleep phases from short EEG segments known as epochs. Researchers have delved into various aspects of this domain, with a specific emphasis on predicting human behavior and enhancing road safety within VANETs. Cloudin *et al.* [78] provide valuable insights into Driver Behavior Detection in the VANET environment, shedding light on predicting human behavior to improve road safety. Abbood *et al.* [79] explore the feasibility of detecting fatigue by considering a cluster of signs, including pupil reaction, visual attention, steering response, and EEG signals, aiming for early fatigue prevention. Detecting microsleep episodes using a single EEG electrode involves intricate signal processing, and conventional artifact removal techniques [80]. Machine learning, including advanced black-box methods [81–85], holds significant potential in this domain, but the choice of model architecture has a substantial impact on accuracy. Phan *et al.* [86] propose a trans-

fer learning strategy to automate sleep staging using single-night data, mitigating overfitting. Seo *et al.* [87] introduce Temporal Context Networks for Automatic Sleep Scoring, emphasizing the importance of intra and inter-relationships in EEG analysis. Saeed *et al.* [88] leverage contrastive loss on Multi-Sensor Networks and Wavelet Transforms, while Phan *et al.* [89] develop a deep transfer learning framework for automated sleep staging. Furthermore, sequence-to-sequence algorithms show promise in sleep staging, as exemplified by MRNet [90]. In [91], researchers utilize raw data and time-frequency images to determine sleep levels and propose a sequence-to-sequence approach. Additionally, Eldele *et al.* [92] advocate for a deep learning architecture employing attention mechanisms for sleep stage categorization, particularly with single-channel EEG inputs. These advancements collectively contribute to the evolution of sleep analysis and its potential applications in diverse contexts.

2.3 Traffic Congestion Management in VANET Environments

The implementation of smart eco-traffic signals represents a promising avenue for enhancing the efficiency and sustainability of transportation systems [93]. Unlike traditional traffic lights, these intelligent eco-traffic signals leverage a range of data sources, including sensors, edge devices, and video cameras, to intelligently manage traffic flow, as highlighted by Lee *et al.* [94]. In smart city transportation, these advanced signals address various transportation challenges, including the prioritization of public transport and facilitating alert broadcasting. This innovative approach contributes to optimizing urban mobility and fostering a more sustainable transportation ecosystem. Intersections are notorious for traffic congestion, and a smart eco-traffic signal system [95], plays a critical role in addressing this issue by reducing vehicle idle time, saving fuel, and improving overall

efficiency. The eco-traffic signal system provides valuable information to vehicles, allowing them to adapt their driving behavior and minimize stops at intersections. Autonomous vehicles can seamlessly communicate with nearby road infrastructure, particularly traffic signals, which, as reported by several researchers [96–100], holds significant potential for reducing greenhouse gas (GHG) emissions. Massar *et al.* [101] demonstrated that eco-traffic signals can reduce the number of stops at intersections, resulting in a 9% reduction in GHG emissions. Rakha *et al.* [102] developed a framework to enhance vehicle fuel efficiency at intersections, highlighting the importance of minimizing fuel consumption and GHG emissions, particularly when vehicles operate at reduced speeds at intersections [103]. Zimmerman *et al.* [104] conducted a comprehensive study on traffic patterns at signalized intersections and found a potential 1.8% reduction in emissions, primarily linked to vehicle acceleration profiles and energy consumption. Signalized intersections in urban areas hold substantial promise for reducing GHG emissions at the network level. Traffic signals predominantly govern intersections today, often promoting stop-and-go driving habits, which are highly inefficient. Poorly optimized traffic signal schedules have been identified as a source of 10% of traffic delays. Researchers have explored coordinating solutions for signal-less intersections, as observed in the works of Dresner and Stone [105] and Du *et al.* [106]. However, the coordination of vehicles has shown significant potential in reducing travel time and fuel consumption in a straightforward manner [107]. To develop a practical signal control strategy for remote intersections, which may feature a mix of manually operated vehicles, connected vehicles, and autonomous vehicles, Yang *et al.* [108] propose a method that predicts the positions of human-driven vehicles and optimizes vehicle trajectories accordingly.

2.4 Perception for Vehicles

In the realm of computer vision tasks, particularly image dehazing, deep learning-based methods have exhibited impressive results. Single Image Dehazing, a common approach, leverages Convolutional Neural Networks (CNNs) to directly transform hazy images into clear ones, capitalizing on extensive hazy-clear image pairs in the training dataset [10, 109–112]. The network's capacity to discern haze-related features and effectively eliminate haze is honed through this learning process. Multi-scale convolutional neural networks (CNNs) have also played a vital role in image dehazing, as different scales have varying impacts on the quality of dehazing [113–118]. The incorporation of multiple scales, either through input variations or within the network's processing, enables the model to capture haze-related information across different scales, resulting in enhanced dehazing performance. Additionally, Mehta *et al.* introduced a mechanism that combines CycleGAN and conditional GANs, guided by hyperspectral data, for image dehazing [119]. Fu *et al.* proposed a GAN-based method utilizing discrete wavelet transform for non-homogeneous hazy environments [120]. Furthermore, the use of pix2pix GAN for dehazing has improved color and detail enhancement in dehazed results [121]. Various other studies have explored single-image dehazing using GANs [122–126]. Nevertheless, the reliance on GANs often poses challenges due to the scarcity of training data for image dehazing and issues like mode collapse, non-convergence, and training complexities. GAN models also demand substantial computational resources and data. Recent transformer-based approaches, such as ViT, Swin, TransRA, DEHRFormer, EHA-Transformer, Self-promer, Semi-UFormer, and Dehazeformer, have shown promise in hazy environments [127–133]. However, transformers and GANs, require significant computational power and resources [134, 135]. In recent years, diffusion models have emerged as a potent tool for computer vision tasks in adverse weather condi-

tions. Yu *et al.* introduced the Denoising Diffusion Probabilistic Model (DDPM) and applied it within a physics-aware framework for image dehazing [136]. Similarly, Wang *et al.* harnessed conditional diffusion models to enhance the dehazing process [137]. Luo *et al.* proposed a U-Net-based latent diffusion model tailored for image dehazing tasks [138]. Özdenizci and Legenstein devised a patch-based image restoration algorithm based on denoising diffusion probabilistic models, offering a novel approach to image dehazing challenges [139].

Chapter 3

Towards Resilient Security Provisioning in VANETs for ADAS-Equipped Vehicles

3.1 Introduction

The evolution of technology in areas such as Internet of Things (IoT), edge and cloud computing, and the advent of 5G have spurred the rapid development of Intelligent Transportation Systems (ITS), exemplified by Vehicular ad hoc networks (VANETs) [140, 141]. VANETs, enabled by Vehicle to Everything (V2X) communication, serve to enhance the safety, sustainability, and efficiency of the network, facilitating communication between vehicular nodes [142, 143]. V2X communication encompasses various modes, including Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Sensor (V2S), and Vehicle to Pedestrian (V2P), with the proliferation of these links expanding the attack surface for potential cyber-threats. These threats encompass data integrity, verification, user privacy, and network availability, among others [144, 145]. It is imperative to underscore

that road accidents claim the lives of approximately 1.3 million people and cause injury to 8 million individuals annually [41, 42, 142]. Misinformation and network unavailability within VANETs can have dire consequences, necessitating the proposition of robust solutions to detect, analyze, mitigate, and prevent cyber-threats.

The considerable volume of data generated and shared within VANETs presents challenges in terms of effective management and analysis of big data. Deep learning has emerged as a promising solution, leveraging the success of recent deep learning-based intrusion detection systems in identifying known vehicular misbehaviors [47, 52]. Furthermore, recent studies have explored efficient resource management solutions for edge servers, mitigating the inherent latency and cost limitations associated with cloud-based alternatives [146]. However, one significant drawback of existing intelligent security measures lies in their reliance on a limited set of known cyber-threats. The emergence of new threats necessitates in-depth analysis, data collection, and retraining of Deep Learning-based solutions to maintain up-to-date intrusion detection capabilities. The long-term maintenance of such solutions is not only expensive but also time-consuming, highlighting a significant shortcoming in state-of-the-art approaches. Thus, it becomes crucial to develop systems capable of efficiently identifying new forms of misbehavior, addressing the evolving threat landscape.

In ADAS-enabled vehicles, a diverse array of sensors such as Cameras, Lidar, Global Positioning System (GPS), and numerous others are deployed to facilitate critical functionalities. These sensors establish communication with one another via the Controller Area Network (CAN) Bus. Additionally, Electronic Control Units (ECUs), which are compact yet integral devices within the vehicle, are entrusted with the management of specific functions, spanning from core tasks like engine and power steering control to amenities such as power windows and en-

tertainment systems [147]. In the modern automotive landscape, a multitude of ECUs are interconnected through the CAN bus, adopting a decentralized architecture where each node is directly linked to every other node within the network [148]. This design eliminates the necessity for a central controller, granting all nodes on the CAN bus equal priority. An intricate web of systems and components within the vehicle is interwoven through an in-vehicle network. The prominence of the CAN bus in ADAS-enabled vehicles is undeniable, owing to its versatility and wide-ranging applications. However, its security posture is marred by several open challenges, rendering it susceptible to potential attacks. Through the CAN bus network, adversaries can potentially seize control of the vehicle, resulting in hazardous driving conditions jeopardizing the safety of the driver, passengers, and pedestrians. In the most dire circumstances, attackers could manipulate the vehicle to create chaos within the VANET environment. Thus, it is imperative to fortify in-vehicle communication, particularly CAN-based communication, through the detection and mitigation of various attack vectors.

In the realm of VANETs and ADAS, the development of robust security models encounters a recurring challenge: the limited availability of data. The efficacy of deep learning models is intrinsically linked to the quantity of data available for both training and testing. When data is scarce, models often struggle to attain the desired level of performance and may exhibit a lack of robustness. To address these challenges, the application of data augmentation techniques becomes imperative. Data augmentation encompasses the process of generating additional data points by subjecting the existing dataset to a variety of transformations. Through this augmentation, researchers and practitioners can significantly enhance the diversity of the training data, thereby facilitating more effective learning and bolstering the model's robustness. The employment of data augmentation and synthetic dataset generation techniques stands as a pivotal component in the advancement of security in VANET and ADAS applications, enabling the creation of more po-

tent and resilient models that are better equipped to tackle the intricacies of real-world scenarios. The key contributions of this chapter are as follows:

1. Introduced Multi-branch Reconstruction Error (MbRE), a deep learning-based intrusion detection model designed specifically for intelligent edge-based policing within VANETs.
2. MbRE is an edge-deployed, deep learning, and statistics-based IDS designed for the cost-effective and long-term security of VANETs against a wide range of cyber threats, without the need for training on specific threats.
3. The model architecture of MbRE showcases the detection capability on nineteen types of data integrity, frequency, and identity-based misbehaviours, all of which were previously unknown to the IDS.
4. Additionally, MbRE demonstrates its hardware implementation practicality in both cloud and edge environments, such as Google Colaboratory and Jetson Nano.
5. Introduced HybridSecNet, a two-step Intrusion Detection System (IDS) designed explicitly for in-vehicle communication security threats, especially in CAN.
6. HybridSecNet exhibits resource-saving and real-time capabilities due to its efficient design. When the input data is not identified as an attack type in the first step, the second step, responsible for determining the attack type, is not invoked. This feature ensures that the second step is utilized only when necessary. As a result, unnecessary computational resources are conserved, leading to faster and more efficient data processing.
7. Compared to a single-step IDS model for in-vehicle communication attack detection, the proposed HybridSecNet demonstrates improved performance in terms of classification accuracy and robustness, as evidenced by evalua-

tions on various metrics, including Accuracy, Precision, Recall, and F1 score.

8. Furthermore, we introduced a novel GAN method called 'SCAN-GAN' for the generation of synthetic CAN data, demonstrating the adaptability of GANs to work with diverse data and their customizable nature to generate data of a specific nature as needed.

3.2 MbRE Model

This section introduces a MbRE intrusion detection architecture designed specifically for intelligent edge-based policing within VANETs. The MbRE model has been developed with the primary objective of enhancing security within the context of edge scenarios in VANETs. This encompasses safeguarding communication networks and data exchange both between vehicles (V2V) and between vehicles and roadside infrastructure (V2I) within the VANET environment. The MbRE model operates by analyzing network traffic data, encompassing information such as timestamps, message logs, source and destination addresses, and other pertinent network-related features. Through this analysis, the model is capable of continuously monitoring incoming data streams, identifying patterns indicative of potential intrusions or security threats. Once the model is appropriately trained, it can be deployed to actively oversee real-time VANET network traffic, ensuring rapid detection of security breaches. The deployment of an IDS model at the edge within a VANET setting offers several distinct advantages. By leveraging edge computing capabilities, the MbRE model guarantees real-time security monitoring, which is paramount in the time-sensitive VANET domain, thereby fortifying the resilience and dependability of vehicular communication systems.

Deploying an IDS model at the edge within a VANET scenario offers several advantages. Edge devices, such as onboard units (OBUs) in vehicles or roadside units (RSUs), enable local processing of network data, allowing for real-time

analysis of network traffic to detect intrusions and perform other security tasks. Real-time analysis is vital for maintaining the security of VANETs and responding promptly to potential threats. Edge deployment in VANETs reduces latency by eliminating the need to transmit data to centralized servers or the cloud for analysis. This reduction in latency is particularly beneficial for latency-sensitive applications like collision avoidance. Additionally, edge processing minimizes unnecessary data transmission, thus lightening the network's load and minimizing the transfer of non-essential data. Moreover, deploying IDS models at the edge enhances privacy by reducing the volume of sensitive data transmitted over the network. The distributed architecture of edge deployment aligns well with the decentralized nature of VANETs, mitigating the risk of single points of failure and enhancing overall network robustness. However, it's essential to consider the resource constraints of edge devices in VANETs, particularly OBUs, which often have limited computational resources like CPU and memory. Therefore, deep learning models deployed at the edge must be optimized for resource efficiency to ensure effective operation. For model training and updates, centralized training with ample computational resources is common for deep learning models used in intrusion detection. Once trained, these models can be deployed to edge devices. Regular model updates are often necessary to adapt to evolving threats, and these updates must be efficiently distributed to edge devices. Additionally, collaborative edge devices in VANETs can enhance security by sharing information about detected threats or anomalies. For instance, OBUs in close proximity can collaborate, allowing for coordinated responses to potential security threats. Finally, edge devices must exhibit robustness to network disruptions, which are frequent in VANETs due to the dynamic mobility of vehicles. Models should be designed to withstand intermittent connectivity and seamlessly resume operation following temporary outages. These advantages underscore the significance of deploying IDS models at the edge within VANET scenarios. In short a VANET

scenario, deploying deep learning models at the edge for intrusion detection is a powerful approach that leverages the benefits of real-time processing, reduced latency, and distributed architecture.

3.2.1 Dataset and Pre-processing

The MbRE model was developed using the VeReMi Extension dataset for intrusion detection [70]. This dataset was chosen for its robustness and the notable variety of vehicular behaviors it encompasses. The dataset consists of twenty-four V2V and V2I communication folders, each corresponding to 24 hours of simulation. The traceGroundTruth.json files within these folders contain multiple data points, including the vehicles' (X, Y) coordinates of position, velocity, acceleration, and heading, the pseudo-identity used by the vehicle to broadcast the message, as well as the timestamp at which the message is received by the infrastructure.

Due to the absence of acceleration and heading-based misbehaviors, only six data fields were utilized for the purpose of intrusion detection. These fields include timestamps for the F-branch, transformed pseudo-identities for the I-branch, and (X, Y) position and speed coordinates for the M-branch. The data was organized into a dictionary of vehicles labeled with their respective behaviors and converted into sequences of dimensions (20,6) with the F, I, and M branches having dimensions (20,1), (20,1) and (20,4) respectively. The sequences were generated with a window length of 5, meaning that a vehicle with 50 data points would have a total of 6 sequences of data. The first sequence covers data points 1 to 20, the second covers data points 5 to 25, and so on. This windowed sequence generation was implemented to ensure completeness of information within a sequence while minimizing data redundancy. Subsequently, the data was scaled. The first timestamp of each F-branch sequence was subtracted from all timestamps in that sequence to remove the possibility of bias towards exact timestamp values. In a

similar fashion, the pseudo-identities were transformed as a function of the vehicle's actual identity. The positions and velocities in the M-branch were scaled down by factors of 100 and 10, respectively.

3.2.2 MbRE Architecture

An overview of the proposed MbRE IDS is depicted in Figure 3.1. MbRE is a statistical and intelligent IDS designed to identify both known and unknown vehicular misbehaviors by analyzing message frequency, identity, and data, without the requirement for prior training on these behaviors.

3.2.2.1 Multi-branch Sequences

Data broadcast from a vehicle contains three primary pieces of information: the frequency of messages (F), the pseudo-identities assumed for the messages (I) and the position and velocity data (M). These three branches, F-I-M, constitute the foundation of the proposed IDS. The time-series information for each vehicle in the network is structured into sequences and separated into F-I-M branches to facilitate independent analysis.

3.2.2.2 F-I-M Encoding

According to the data, each of the behaviors exhibits misbehavior in one or more of the F-I-M branches. For example, DoS attacks consist of an F-fault, while Sybil and Random Position attacks consist of I and M-faults, respectively. Based on the type of faults present in the broadcast, a vehicle's behavior can be encoded into 3-bit strings of F-I-M, resulting in 8 unique F-I-M categories of vehicular behavior, creating distinct "themes" of vehicular behaviors. Table 3.1 illustrates the eight general F-I-M encodings and the corresponding twenty sample vehicular behaviors considered. As shown in the table, the DoS attack is encoded as 1-0-0,

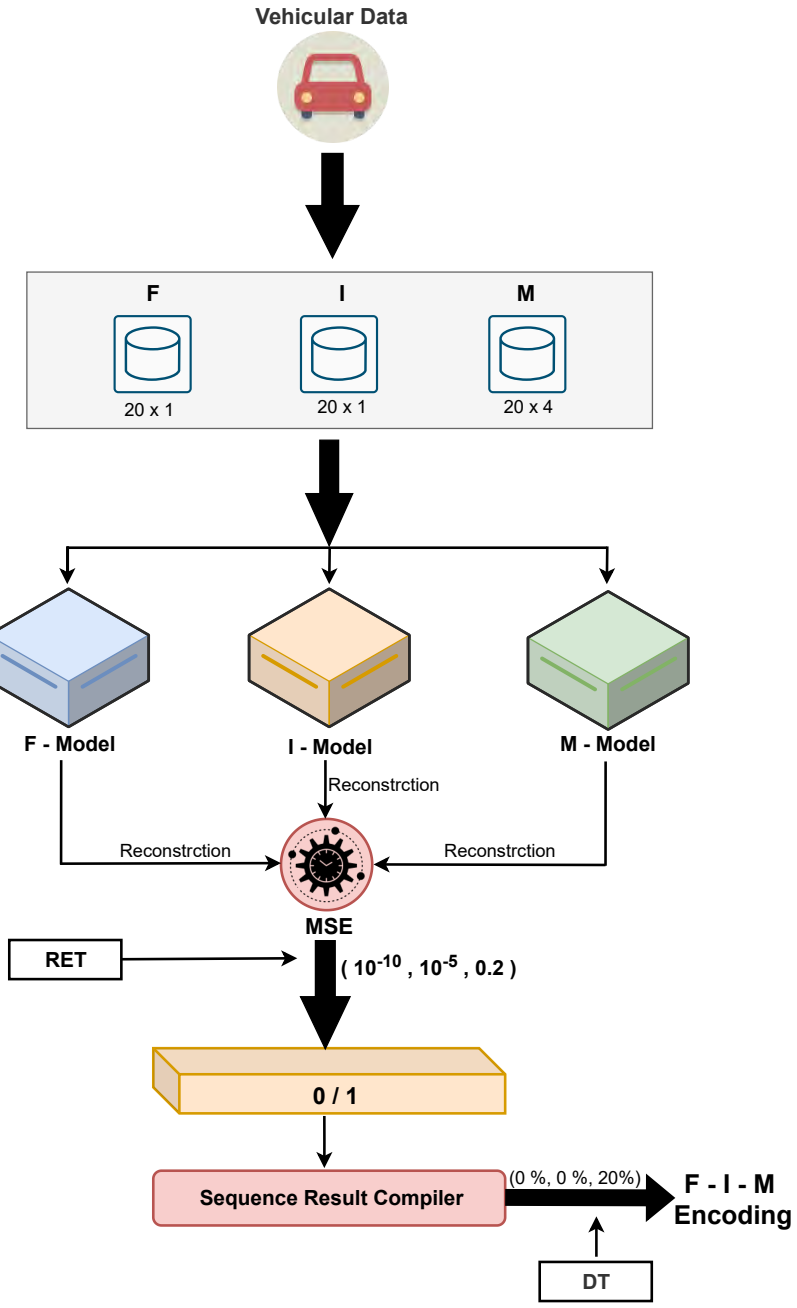


Figure 3.1: Structure of the MbRE IDS

while a more complex attack such as DoS Disruptive Sybil is encoded as 1-1-1 since it involves misbehavior in each of the three data branches. It's important to note that all behaviors in the table, except for normal behavior, are unexplored to the IDS.

S.No.	F-I-M Encoding	Sample Behaviors
1	0-0-0	Normal
2	0-0-1	Constant/Random/Offset position and speed, Eventual stop, Disruptive, Data replay, Delayed messages
3	0-1-0	Sybil
4	0-1-1	Traffic congestion sybil
5	1-0-0	DoS
6	1-0-1	DoS random, DoS disruptive
7	1-1-0	Not available
8	1-1-1	Data replay, DoS random sybil, DoS disruptive sybil

Table 3.1: Sample vehicular behaviors for the eight possible F-I-M encodings (Frequency of messages (F), Pseudo-identities (I) & Motion (M))

3.2.2.3 Reconstruction Error

Reconstruction is a deep learning approach employed to recreate the input provided to a model. In this study, three Convolutional Neural Network (CNN) models were trained to reconstruct the individual F-I-M branches of normal vehicular behavior. The Reconstruction Error (RE) for each reconstructed branch can be assessed as its Mean Square Error (MSE) relative to the original input sequence. Therefore, for a given i^{th} input sequence S_i and the reconstructed output sequence S'_i, the RE_i is calculated as,

$$RE_i = 1/m \sum_{j=1}^{m} (S_{ij} - S'_{ij})^2 \tag{3.1}$$

where m is the dimension of the input sequence vector, S_{ij} is the j^{th} data point in sequence S_i. Therefore, Reconstruction error (RE) provides a measure of the closeness of a reconstructed input sequence to the original one. It's important to note that the three CNN models reconstruct the F-I-M data branches individually. Therefore, a single complete reconstructed sequence of vehicular data consists of three separate Reconstruction error (RE) values, one for each of F, I and M branches.

3.2.2.4 Sequence-level Encoding and Reconstruction Error Threshold

Sequence-level encoding (SE) refers to the predicted F-I-M encoding of an input sequence. This prediction is made using a Reconstruction Error Threshold (RET), which is defined as the minimum error value above which the behavior of a vehicle in a data branch is considered malicious. The RET for each branch is chosen to be slightly larger than the maximum RE of normal behavior in that branch after filtering outliers. The i^{th} reconstructed sequence branch b (where b can be one of F, I, or M) is encoded as 0 or 1 based on its value relative to the respective RET. as per the following equation,

$$SE_{bi} = \begin{cases} 0, & \text{for } RE_{bi} \leq RET_{bi} \\ 1, & \text{for } RE_{bi} > RET_{bi} \end{cases} \tag{3.2}$$

where SE_{bi} is the sequence level encoding of branch b of the ith sequence. On compiling the three branches together the overall vector SE_i can be written as,

$$SE_i = [SE_{Fi}, SE_{Ii}, SE_{Mi}] \tag{3.3}$$

3.2.2.5 Vehicular Encoding and Detection Threshold

In addition to the RET, which is responsible for the sequence-level encoding of vehicular behavior, a Detection Threshold (DT) is employed to evaluate the overall behavior of a vehicle in the network. The DT is defined as the minimum proportion (in percentage) of sequences identified to have errors (using RET) above which the overall behavior of a vehicle is labeled as malicious. For a vehicle with n sequences, the Branch Ratio (BR), representing the ratio of sequence-level mis-

behavior predictions to the total number of sequences, is calculated as,

$$BR_b = 1/n \sum_{i=1}^{n} (SE_{bi}) \tag{3.4}$$

Accordingly, the Vehicular Encoding (VE) for data branch b can be calculated as,

$$VE_b = \begin{cases} 0, & \text{for } BR_b \leq DT_b \\ 1, & \text{for } BR_b > DT_b \end{cases} \tag{3.5}$$

where DT_b is the DT of branch b. Similar to the compiled RET, the overall VE can be written in a concise form as,

$$VE = [VE_{Fi}, VE_{Ii}, VE_{Mi}] \tag{3.6}$$

3.2.2.6 Reconstruction Error and Detection Thresholds

After conducting various experiments, an RET of $[10^{-10}, 10^{-5}, 0.2]$ and a DT of $[0\%, 0\%, 0.2\%]$ were chosen to achieve the highest possible final recall scores. For F and I reconstruction, no outliers were observed in normal behavior; therefore, thresholds slightly higher than the maximum reconstruction errors were selected for these data branches. In contrast, it was noted that the M-model reconstruction had approximately 5% of data points as outliers, so an error threshold of 0.2 was set for the M-branch. Based on the severity of misbehavior in the data branches, the DT for F and I branches was fixed at 0% to classify vehicles displaying any misbehavior in these categories as malicious. Meanwhile, the DT for the M-branch was set at 20% to accommodate the expected high variance and possible values of position and velocity.

3.2.3 MbRE training and Hyperparameters

Three CNN reconstruction models were employed for the F-I-M data branches, all of which were trained to reconstruct normal data. The F and I models consisted of one Conv1D layer with 32 filters and a kernel size of 2, followed by a MaxPool1D layer with a pooling size of 2. Subsequently, there was a Flatten layer, and two Dense layers with 50 neurons and 20 output neurons. The M-model comprised of two Conv2D layers with 64 (2x1) kernels and 32 (2x1) kernels, followed by a MaxPool2D layer with a size of (2x1). It also had a Flatten layer and two Dense layers with 50 neurons and 80 output neurons. The F and I models had 20 output neurons each to reconstruct the (20,1) dimension F and I-branches, while the M-model had 80 output neurons to reconstruct the (20,4) M-branch sequences in a (80,1) format. All models used the ReLU activation function and were optimized with the Adam optimizer at a learning rate of 0.001. It's worth noting that experiments with Stochastic Gradient Descent (SGD), another popular optimizer choice, yielded less favorable results. The F and I models were trained for 10 epochs, while the M-model was trained for 200 epochs due to the inherent complexity of the M-branch data. The proposed reconstruction models were trained using 10,000 normal behavior sequences. After training, sequences from 300 vehicles representing each of the 20 behavior types listed in Table 3.2 were reconstructed for evaluation.

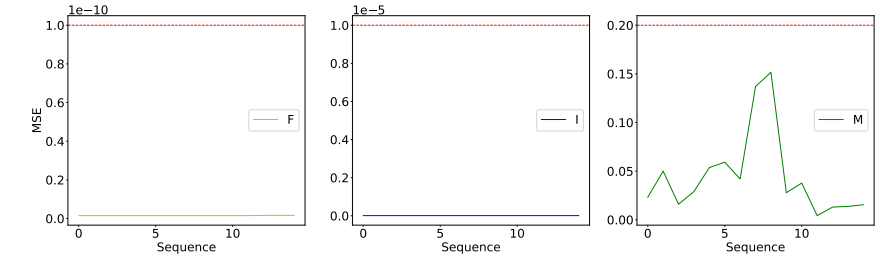
3.2.4 Results

The IDS classifies each branch into two classes (0 or 1), making precision, recall, and accuracy interchangeable. For MbRE, evaluation is primarily based on the recall parameter. Table 3.2 presents the recall scores for test samples of 300 vehicles from each of the twenty vehicular behaviors included in the VeReMi Extension dataset. The F and I models perform exceptionally well, achieving 100% recall for

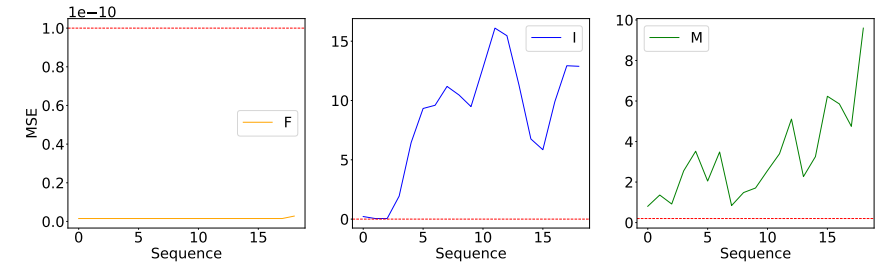
S.No.	Behavior	Encoding			Recall		
		F	I	M	F	I	M
0	Normal	0	0	0	100	100	98
1	Constant position	0	0	1	100	100	42.5
2	Constant position offset	0	0	1	100	100	5.5
3	Random position	0	0	1	100	100	100
4	Random position offset	0	0	1	100	100	8
5	Constant speed	0	0	1	100	100	94.5
6	Constant speed offset	0	0	1	100	100	42
7	Random speed	0	0	1	100	100	100
8	Random speed offset	0	0	1	100	100	100
9	Eventual stop	0	0	1	100	100	9
10	Disruptive	0	0	1	100	100	98.5
11	Data replay	0	0	1	100	100	98.5
12	Delayed messages	0	0	1	100	100	8.5
13	DoS	1	0	0	100	100	98
14	DoS random	1	0	1	100	100	100
15	DoS disrptive	1	0	1	100	100	99.5
16	Data replay sybil	1	1	1	100	99.5	98.5
17	Traffic congestion sybil	0	1	1	100	98.5	97.5
18	DoS random sybil	1	1	1	100	100	100
19	DoS disruptive sybil	1	1	1	100	99	99

Table 3.2: F-I-M encoding (Frequency of messages (F), Pseudo-identities (I) & Motion (M)) and branch recall results of MbRE for the 20 vehicular behaviours including 19 attacks and one normal behaviour in the VeReMi Extension dataset.

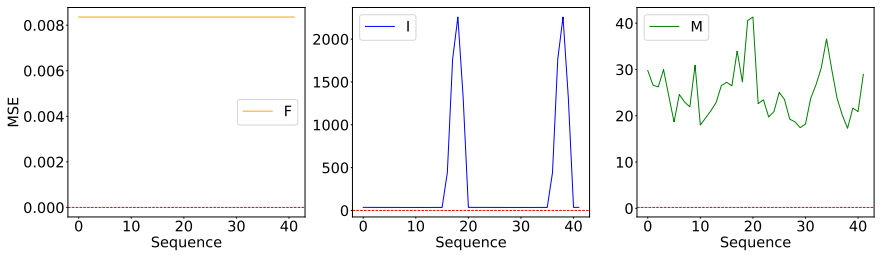
the F-model and 98.5-100% recall for the I-model. The M-model achieves recall scores in the range of 5.5%-100%, with the majority of behaviors classified with recall scores higher than 94.5%. Figure 3.2 visually represents the IDS' error generation and RET (dashed red line) for all sequences of one sample vehicle for normal (type 0) and complex behavior types (types 16, 17, 18, and 19). The figure shows that the RE of the F branch remains nearly constant across all sequences for a given vehicle, reflecting the simplicity of F-based misbehaviour. However, if there were varying message frequencies for a single vehicle across different sequences, the relative change in RE would be observable. Examples of this variance in RE results for different I and M branch sequences of the same vehicle demonstrate the



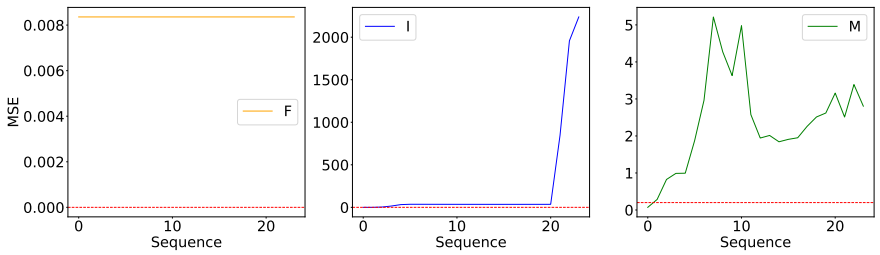
(a) Normal



(b) Traffic congestion sybil



(c) DoS random sybil



(d) DoS disruptive sybil

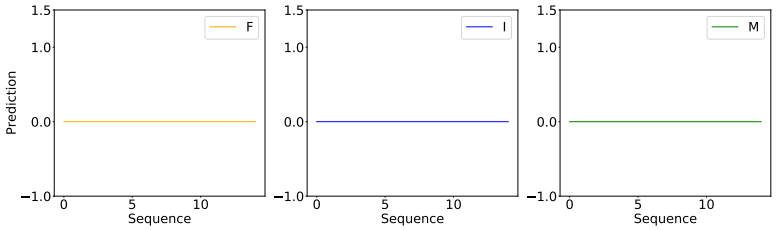
Figure 3.2: F-I-M (Frequency of messages (F), Pseudo-identities (I) & Motion (M)) reconstruction error plots for normal behavior and four complex misbehaviors for sample vehicles.

high variance in the I and M data present in the dataset. These results showcase the IDS's ability to detect different variations in data for a specific behavior type.

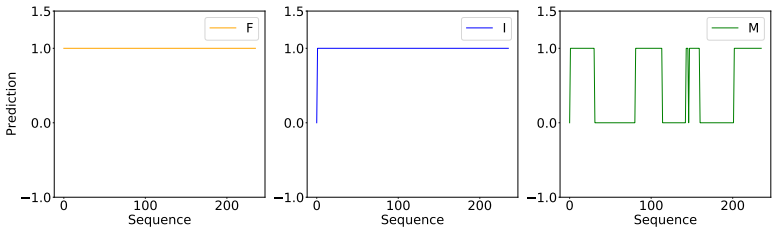
Figure 3.3 & Figure 3.4 displays the classification of vehicular sequence reconstruction errors from the previous figure into the respective sequence-wise encoding

using the RET. Sequences with errors above the RET are classified as 1 (misbehavior), while those below it are classified as 0 (normal). While the IDS may occasionally misclassify some sequences, it can still correctly identify the vehicular behavior type using the DT. For instance, in the case of data replay sybil, the M branch shows spikes in 0 and 1 predictions due to the high variance in motion data. However, even when some sequences are predicted as 0, the overall vehicle behavior is still labeled as having an M-fault. This demonstrates the IDS's reliability in detecting vehicles that may intermittently exhibit misbehavior. Additionally, the results illustrate the IDS's ability to detect combinations of misbehaviors without the need for explicit training, indicating its potential to detect unknown behaviors that deviate from the norm.

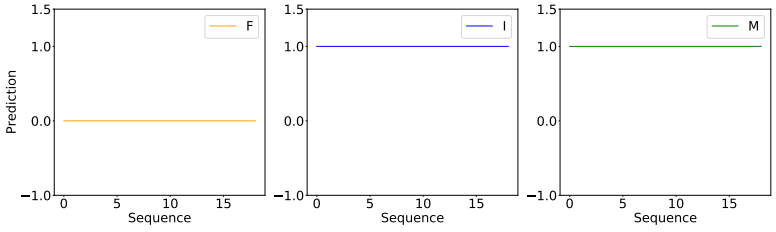
Furthermore, Figure 3.5 presents the error and prediction performance of MbRE for sample vehicles from behavior types 9 (eventual stop) and 12 (delayed messages). The performance evaluation of the MbRE IDS reveals several key insights. The IDS exhibits a robust capability to detect various frequency-based faults commonly encountered in real-world VANETs. Specifically, it effectively identifies Sybil attacks of types 16, 18, and 19, even though it was noted that incorrectly labeled sample vehicles displaying such behavior were more predominant in cases where these vehicles spent an extended duration within the simulation network. This observation underscores the IDS's proficiency in detecting unnatural identity behavior. Nevertheless, certain limitations were evident in the IDS's ability to identify misbehaviors of types 1 (constant position), 2 (constant position offset), 4 (random position offset), 6 (constant speed offset), 9 (eventual stop), and 12 (delayed messages). This limitation primarily stems from the fact that misbehaviors of types 1, 2, 4, and 6 do not generate reconstruction errors of sufficient magnitude to be discerned as anomalies, as they closely resemble normal vehicular behavior. Types 9 and 12, however, represent special cases due to their distinct nature. "Eventual stop" misbehavior emulates the behavior of a vehicle that ini-



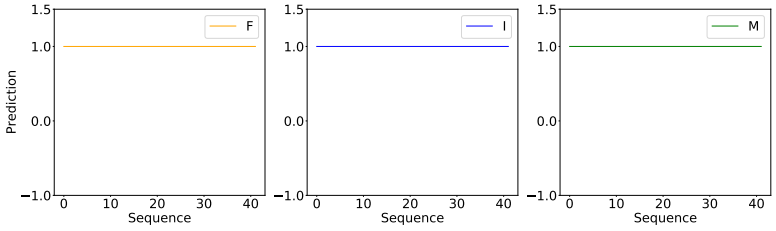
(a) Normal



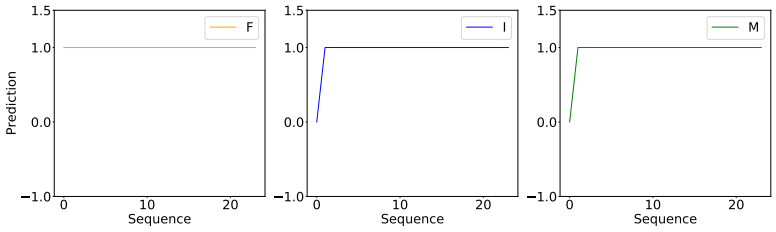
(b) Data replay sybil



(c) Traffic congestion sybil



(d) DoS random sybil



(e) DoS disruptive sybil

Figure 3.3: F-I-M (Frequency of messages (F), Pseudo-identities (I) & Motion (M)) encoding plots for the previously shown reconstruction errors.

tially broadcasts regular vehicular data but comes to an abrupt halt, leading to spikes in the Mean Error (M-error) as depicted in Figure 3.5. A small number of

Device	Setup	Prediction			Encoding	Total
		F	I	M		
Google Colab	0.15	56.55	50.91	50.77	0.21	158.59
Jetson Nano	0.07	171.91	171.68	167.68	0.55	511.89

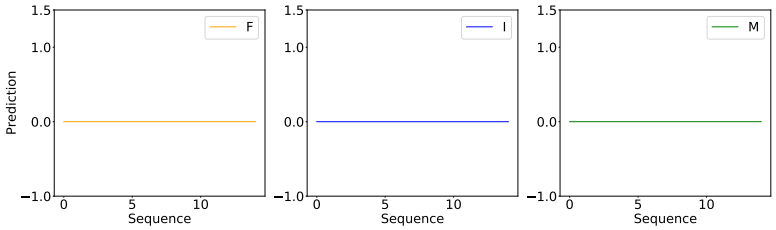
Table 3.3: Run-time results of MbRE on Google Colab (CPU) and Jetson Nano.

points exhibiting such characteristics may elude detection through the Decision Tree (DT). However, these peculiar cases can be identified through manual examination of vehicular behavior plots following an in-depth study of specific error thresholds for each behavior.

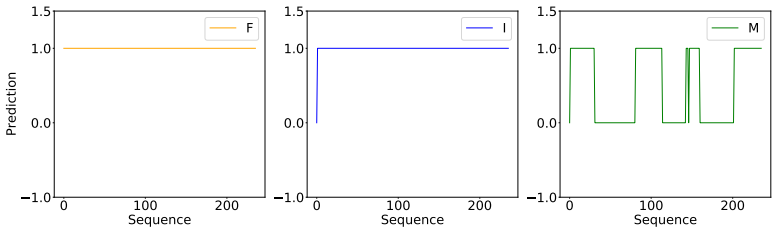
In terms of runtime performance, Table 3.3 presents the average execution times in milliseconds for various components of the proposed IDS, assessed in two distinct environments: the cloud-based Google Colaboratory (CPU) and the Jetson Nano, offering a more real-world perspective. The IDS demonstrates the capability to perform sequence generation, prediction, and encoding for each vehicle in an average time of 511.89 milliseconds on the Jetson Nano when employing sequential prediction. Notably, the execution time can be significantly reduced to approximately 180 milliseconds per vehicle through parallel prediction. On Google Colaboratory, the IDS exhibits an even swifter performance, with an overall series average runtime of 158.59 milliseconds. By utilizing parallel processing, this time can be further reduced to 60 milliseconds. These results underscore the efficacy of the MbRE IDS in executing its operations within practical timeframes, rendering it well-suited for deployment in real-world VANET scenarios.

3.3 HybridSecNet Model

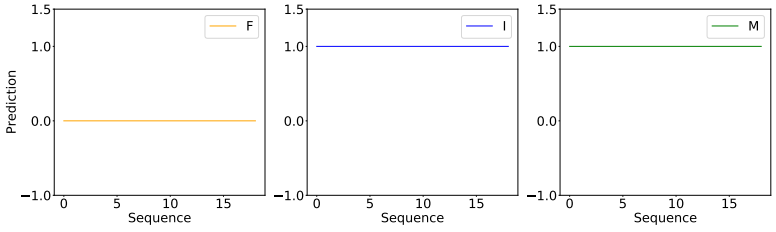
While CAN stands out as a widely adopted in-vehicle communication protocol, offering distinct advantages over alternatives like LIN, FlexRay, and Ethernet, it is important to acknowledge that it also presents certain vulnerabilities. CAN is extensively employed in various automobiles, including those equipped with



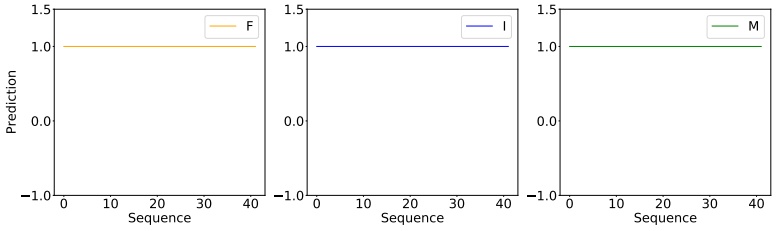
(a) Normal



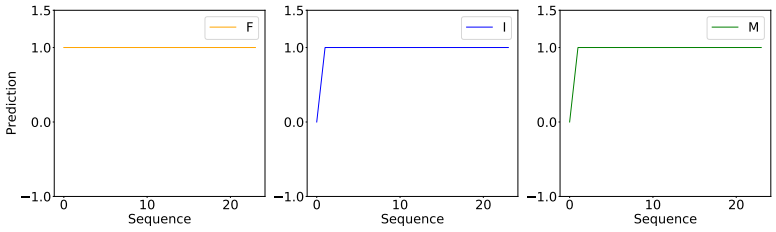
(b) Data replay sybil



(c) Traffic congestion sybil



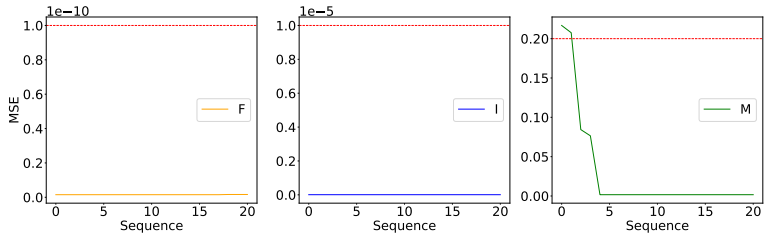
(d) DoS random sybil



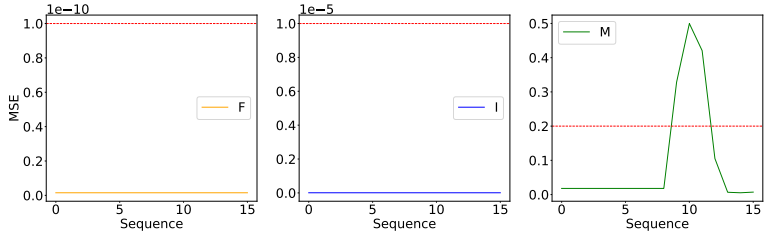
(e) DoS disruptive sybil

Figure 3.4: F-I-M (Frequency of messages (F), Pseudo-identities (I) & Motion (M)) encoding plots for the previously shown reconstruction errors.

ADAS, due to its efficiency and reliability. However, these advantages are tempered by the fact that CAN exposes an attack surface susceptible to multiple threat



(a) Eventual stop



(b) Delayed messages

Figure 3.5: F-I-M (Frequency of messages (F), Pseudo-identities (I) & Motion (M)) reconstruction error plots for special case behaviors.

vectors, including Denial of Service (DoS), Fuzzy attacks, Gear attacks, and RPM attacks. The implications of such attacks are far-reaching, as they can compromise the functionality of ADAS-enabled vehicles, potentially leading to uncontrollable situations, accidents, loss of life, and chaos in the VANET environment. Therefore, the imperative lies in the development of mechanisms capable of detecting and mitigating in-vehicle communication attacks.

Furthermore, it is essential to note that the CAN bus serves as the central control unit within a vehicle. Any security mechanism introduced should not impose an undue burden on the CAN bus communication between different points, nor should it be resource-intensive. In this section, we introduce the HybridSecNet model, specifically engineered to address the detection of threats in in-vehicle communication, with a particular focus on the CAN. This model is designed to strike a balance between robust threat detection and resource efficiency, providing a comprehensive solution for securing vehicular communication in ADAS-enabled vehicles.

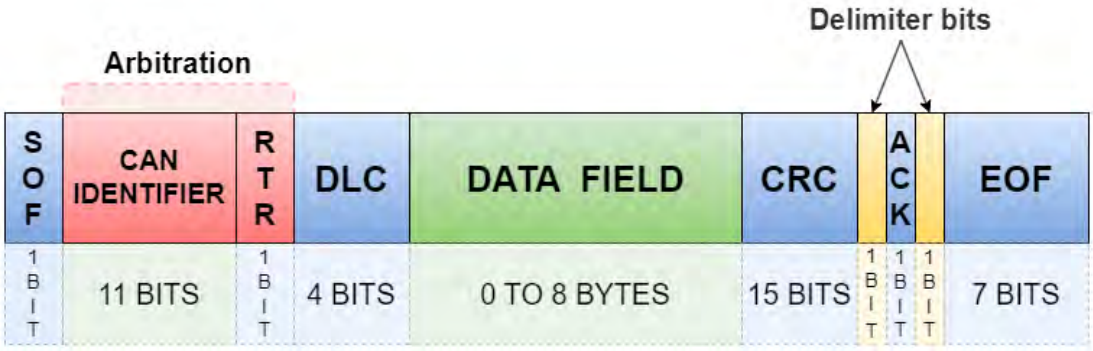


Figure 3.6: CAN format

3.3.1 Dataset and Pre-Processing

The HybridSecNet model was trained and evaluated using the Car Hacking Dataset [149]. This dataset was curated by capturing CAN traffic from an On-Board Diagnostics-II (OBD-II) port of an actual vehicle while subjecting it to message injection attacks. Each dataset within this collection encompasses 300 instances of message injection intrusions, each intrusion spanning a duration of 3 to 5 seconds. The CAN communication data in each dataset covers a timeframe of 30 to 40 minutes. The dataset's data fields follow the structure depicted in Figure 3.6. The timestamp field records time in seconds, the CAN ID represents the hexadecimal identification of a CAN message (e.g., 043f), DL denotes the number of data bytes constituting the CAN frame (ranging from 0 to 8), DATA[0] contains the data value (byte) of the frame, and the Flag field indicates whether the message is injected (denoted as 'T') or a regular message (denoted as 'R'). For training the HybridSecNet model, the initial raw CAN bus dataset, presented in hexadecimal format, requires conversion into decimal format. Within the label column, 'R' is substituted with 0 to denote normal messages, while 'T' is replaced with 1 to signify attack or malicious messages. From the twelve available features in the dataset, ten features were selected for utilization: CAN ID, Data [D0-D7], and the Label column. Each byte of CAN data is allocated to a distinct column (D0-D7).

The Timestamp element was omitted from the feature set as the analysis does not rely on time intervals for intrusion detection. Data standardization was applied to all data fields, and the label column was separated as the output variable. The dataset encompasses four distinct attack types, which include Denial of Service (DoS) [150–152], Fuzzy Attack [153], Gear Attack [149, 154] and RPM Attack [149, 155] as follows,

1. Denial of Service (DoS) Attack [150–152]: The Denial of Service (DoS) attack is distinguished by the deliberate transmission of high-priority messages, resulting in disruptions in the transmission of lower-priority signals. This form of attack is geared towards elevating bus utilization, inducing congestion, and introducing delays in message transmission. These disruptions can significantly impede the arbitration process, leading to adverse consequences within the CAN communication system.

2. Fuzzy Attack [153]:A fuzzy attack, commonly referred to as a bit flipping attack, entails the manipulation of specific bits within a CAN message. These attacks can be challenging to detect since the altered data is typically in close proximity to valid data, giving the impression of a minor error rather than an intentional attack. Fuzzy attacks have the potential to corrupt message content, including identifiers, which can affect the arbitration process and result in incorrect message prioritization.

3. Gear Attack [149, 154]: The gear attack specifically targets messages related to the vehicle's transmission or gear-shifting system. By manipulating these messages, the attack aims to induce abnormal behaviour or complete failure in the targeted Electronic Control Units (ECUs). Such attacks can lead to hazardous situations, including abrupt gear changes, loss of vehicle control, or mechanical damage, posing significant risks to both the driver and the vehicle's operation.

4. RPM Attack [149, 155]: The RPM attack focuses on manipulating messages exchanged between Electronic Control Units (ECUs) responsible for reading RPM sensor data and displaying it on the vehicle's instrument cluster. By tampering with these messages, the attack aims to deceive or disrupt the RPM monitoring system, leading to inaccurate RPM readings being displayed to the driver. This can cause confusion and misinterpretation of the vehicle's engine speed, potentially compromising the driver's ability to make informed driving decisions and posing safety risks on the road.

3.3.2 Intrusion Detection Architecture

The architecture of HybridSecNet, a two-step Intrusion Detection System designed for the CAN, is illustrated in Figure 3.7. Comprising two distinct stages, the HybridSecNet IDS plays a vital role in the comprehensive intrusion detection process. In the first stage, a Long Short-Term Memory (LSTM) based architecture is employed. This LSTM-based classifier is responsible for discerning the presence or absence of attacks within the CAN network. It achieves this by analyzing network data and identifying patterns that signal potential intrusions. When an attack is detected in the initial stage, the second stage of HybridSecNet is activated. This stage employs a Convolutional Neural Network (CNN) based multiclass classifier, prioritizing precision. Its role is to accurately predict the specific type of attack encountered, categorizing the detected attack into one of four predefined classes. The combination of the LSTM-based classifier and the CNN-based multiclass classifier ensures effective attack identification and classification, enhancing the overall security of the CAN network while mitigating potential threats.

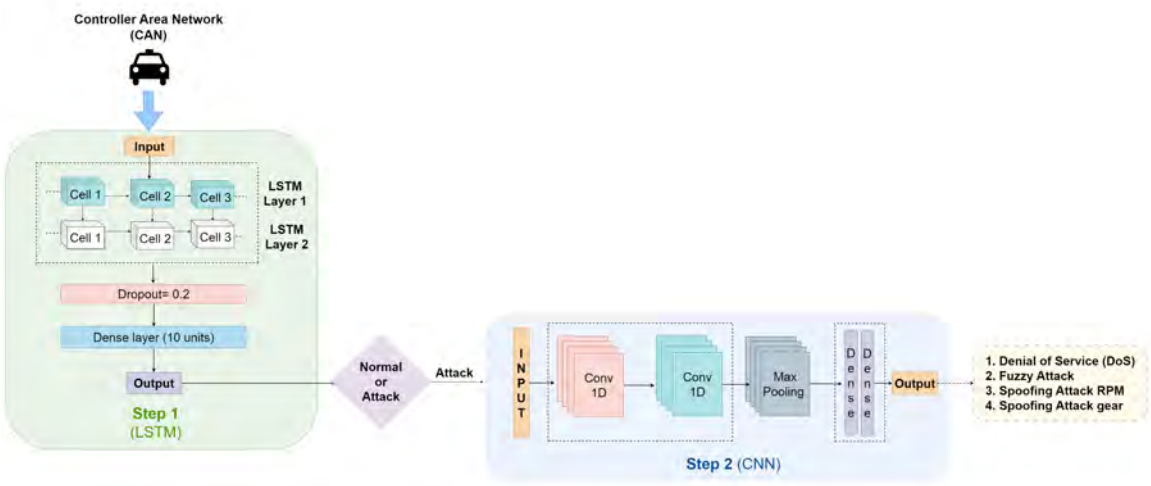


Figure 3.7: Proposed two-step architecture

3.3.2.1 Step 1: Long Short-Term Memory (LSTM) based classifier

In the first step of the HybridSecNet model, an LSTM-based model was developed as a classifier to differentiate between normal and attacked behaviors within the data. The decision to employ LSTM was motivated by its capability to overcome the limitations of conventional Recurrent Neural Networks (RNNs) in capturing long-term relationships within time series data [156]. The LSTM model comprises a cell state that functions as memory, retaining information from past instances, and a "forget gate" that determines which memories can be disregarded by the cell state. These mechanisms enable the preservation of long-term dependencies between past and future instances, effectively addressing the vanishing gradient problem encountered in traditional RNNs. By leveraging the LSTM model, IDS can effectively detect various attack types, whether pattern-based or frequency-based, by capturing long-term temporal trends in the attack-free data and discerning correlations among different variables in the in-vehicle network data. For training the LSTM model, the input data undergoes preprocessing using a technique called window sliding. A window size of 'n' is selected, where the initial 'n' rows from the input and output matrices are utilized to form the first entry in the transformed input and output matrices. This process is iteratively applied to

populate subsequent entries in the transformed matrices. We opted for a window size of 10 when building the model. Figure 3.8 represents the proposed LSTM-based step one architecture. Following data preprocessing, the LSTM classifier

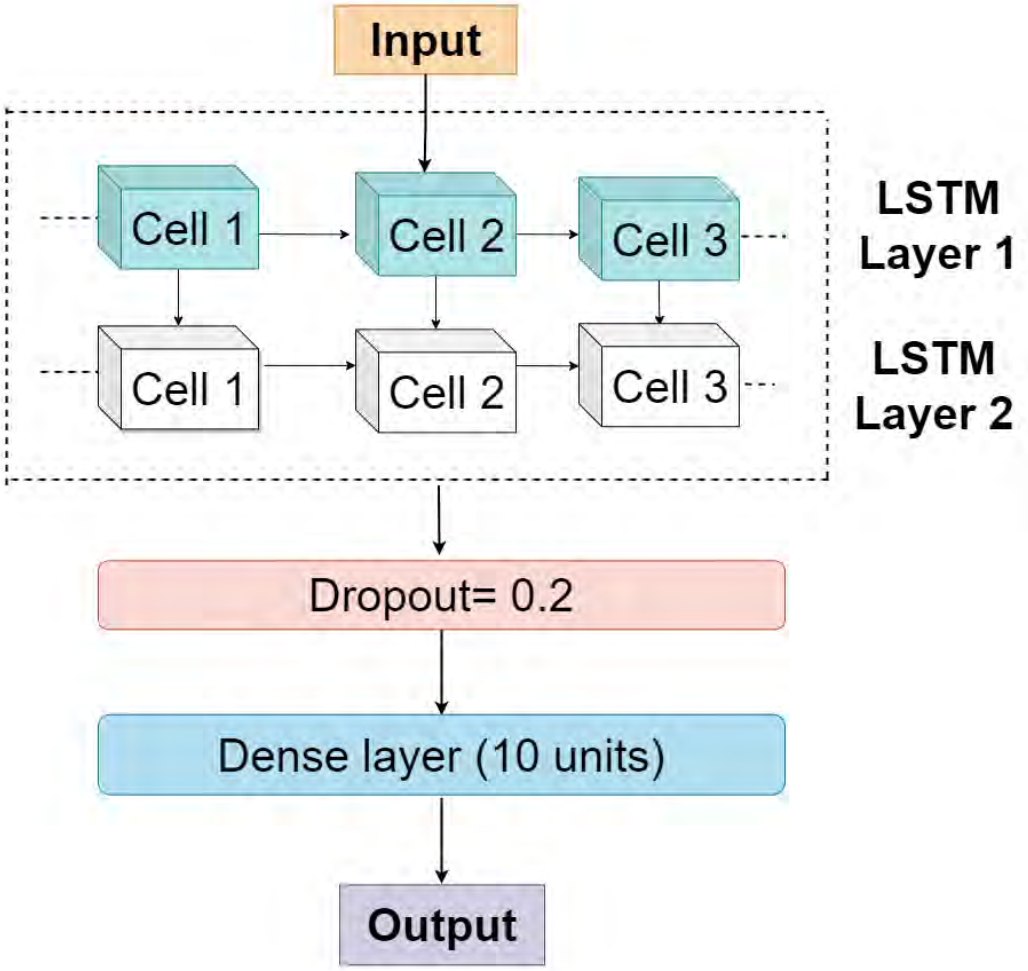


Figure 3.8: Step one architecture (Long Short-Term Memory (LSTM))

is trained using a double-stacked LSTM architecture, comprising 64 units in the first layer and 32 units in the second layer. Through empirical evaluation, a batch size of 64 was found to yield superior detection accuracy. The final layer of the model is a dense layer with a shape of [n*1], receiving input from the preceding LSTM layer. The selection of hyperparameter values was a result of meticulous experimentation. The training process involved the use of the Adam optimizer

Algorithm 1 Transform Predicted Data

Set threshold $t = 0.5$
for each element y in $y_{\text{predicted}}$ **do**
    if $y > t$ **then**
        $y \leftarrow 1$
    else
        $y \leftarrow 0$
    end if
end for
Reshape $y_{\text{predicted}}$ into a 2D array with 10 rows:
    $f =$ number of elements in $y_{\text{predicted}}/10$
    $arr = \text{reshape}(y_{\text{predicted}}, (10, f))$
Initialize a new array arr_{new} with zeros, with a length of $f + 10$:
    $arr_{\text{new}} = \text{create\_array}(f + 10)$
    $\text{fill}(arr_{\text{new}}, 0)$
Get the actual array of predicted values for one-to-one testing:
for j in range(0, columns(arr) - 1) **do**
    for i in range(0, 9) **do**
        $k = i + j$
        $arr_{\text{new}}[k + 1] = arr[i][j] + arr_{\text{new}}[k + 1]$
    end for
end for
Each element of arr_{new} has a value between 0 and 10.
    Set a second threshold:
for i in range(0, size(arr_{new}) - 1) **do**
    if $arr_{\text{new}}[i] > 4$ **then**
        $arr_{\text{new}}[i] = 1$
    else
        $arr_{\text{new}}[i] = 0$
    end if
end for

with a learning rate of 0.0001, and the mean squared error (MSE) served as the loss function, with the output activation function set as a sigmoid. The training spanned 10 epochs. Once the trained model generates predictions on the transformed dataset, an inverse transformation is applied to revert the results to their original shape. Algorithm 1 is utilized for this inverse transformation, followed by the establishment of an appropriate threshold to optimize performance evaluation. When obtaining predictions from the trained HybridSecNet model, direct comparison between predicted values and desired values is not straightforward

due to the window-sliding approach employed for the input data, as previously explained. This approach results in multiple instances of outputs corresponding to each input. To address this challenge, Algorithm 1 consolidates these multiple output instances into a single instance. By establishing a suitable threshold, a single output instance is obtained, enabling a one-to-one comparison between predicted values and actual values for the evaluation of HybridSecNet's performance.

3.3.2.2 Step 2: Attack Categorization using CNNs

Upon detecting malicious or abnormal data in the first step using the LSTM classifier, the data is forwarded to the second step for further analysis and categorization. In the second step, the attacks identified by the first step are classified into four distinct types: DoS, Fuzzy, Gear, and RPM. Given that the first step has already processed the attack-free messages, the primary goal of the second step is to build a classifier capable of distinguishing between these four different attack types. To achieve this, we employed a CNN-based multiclass classifier, as the first step has already considered the temporal correlations and focused on analyzing the spatial features of the attacks. Figure 3.9 visually represents the proposed architecture for step two of HybridSecNet, which is based on Convolutional Neural Networks (CNN). Convolutional Neural Networks (CNNs) are

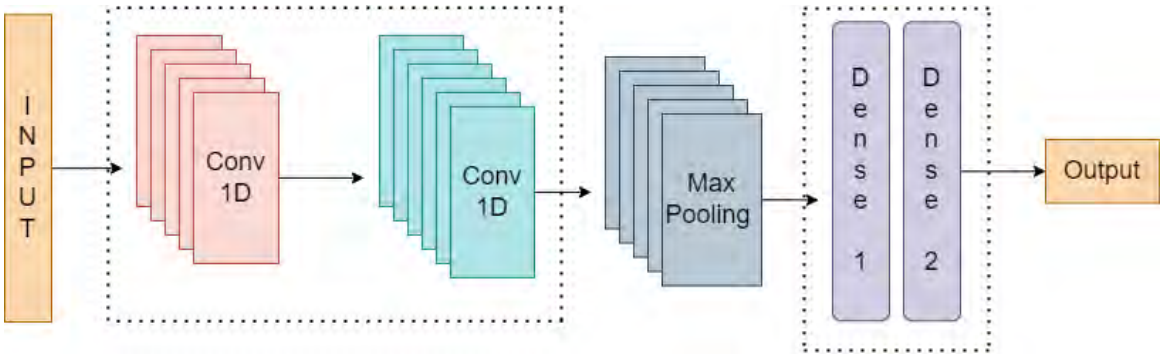


Figure 3.9: Step two architecture (CNN)

particularly well-suited for the detection of intricate patterns and correlations that

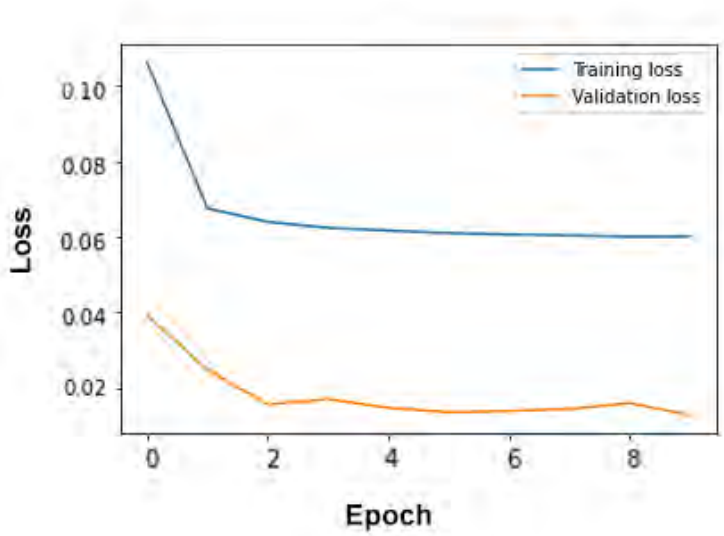
may elude traditional machine-learning approaches. Through convolutional operations and pooling layers, CNNs acquire hierarchical representations that capture significant spatial elements contributing to overall patterns and relationships within the dataset. To facilitate the feature generation process for HybridSecNet, a function was implemented to create features used as input for the CNN component of the model. This function takes the preprocessed data as input, along with the corresponding labels and parameters defining the size and shape of the features. Initially, the function converts the hexadecimal values in the dataset to integers and scales specific columns using a MinMaxScaler to ensure the data is in an appropriate format for further processing. Subsequently, the function iterates over the dataset, selecting consecutive rows as a feature and storing these features in a list. Finally, the function creates a data frame containing the generated features and their respective labels. The CNN model is composed of Sequential, Conv1D, and Dense layers. HybridSecNet accepts the 1D features generated by our function as input data. The model extracts relevant features using two Conv1D layers with Rectified Linear Unit (ReLU) activation, followed by a MaxPooling1D layer for downsampling. The Flatten layer is employed to prepare the data for the subsequent fully connected layers. The model includes two Dense layers with ReLU and softmax activations, respectively, for learning and classifying high-level features. Training of the model is conducted using the Adam optimizer, categorical cross-entropy loss function, and categorical accuracy as the evaluation metric.

3.3.3 Results

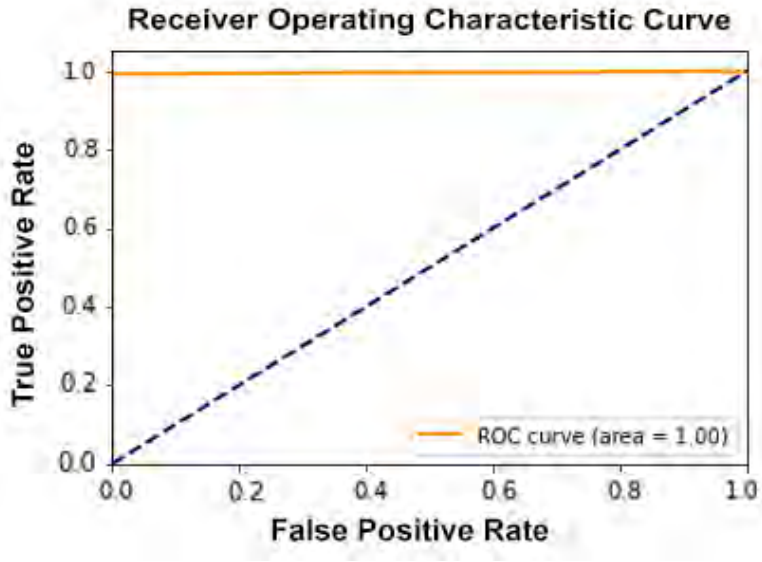
The initial step of HybridSecNet involves an LSTM-based binary classifier responsible for classifying CAN messages into either normal or attack instances. Upon successful detection of an attack, the second step employs a CNN-based "4-attack classifier" to further categorize the attack into one of the four attack types. Conse-

quently, HybridSecNet is a two-step model that not only predicts the presence of an attack but also identifies the specific type of attack if detected. To evaluate the performance of each step, standard evaluation metrics such as accuracy, precision, recall, and F1 score are employed. These metrics offer insights into the model's effectiveness in accurately detecting and classifying attacks. The use of a binary classifier in the first step facilitates training the second step on a dataset containing only attack instances. This approach enhances the performance of the subsequent multi-class classifier by focusing exclusively on attack patterns and eliminating the complexities introduced by normal messages. Consequently, the second step, the CNN-based multi-class classifier, benefits from an improved dataset, leading to enhanced classification accuracy and robustness. The numerical and graphical results of our proposed 2-Step IDS are presented to demonstrate its effectiveness in safeguarding ADAS vehicles from various attack types. Figure 3.10a illustrates the variation of losses with epochs for the first step of HybridSecNet. Additionally, Graph 3.10b depicts the Receiver Operating Characteristic (ROC) graph for step one, showcasing the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. An area under the ROC curve equal to '1' indicates nearly perfect classification. For the second step of HybridSecNet, Figure 3.11 illustrates the variation of Accuracy, F1-score, Loss, Precision, and Recall with epochs. The overall performance of the Hybrid Two-Step LSTM-CNN Model is determined by combining the scores from both the LSTM binary classifier and the CNN multi-class classifier. A comparative analysis is conducted between two variations of the proposed intrusion detection system: a CNN-based single-step classifier and a CNN with LSTM-based two-step classifier. The comparison aims to assess the impact of the additional LSTM-based classifier in terms of classification accuracy and system efficiency. Table 3.4 presents the results of this comparison.

The findings clearly demonstrate that incorporating the extra step in our IDS, uti-



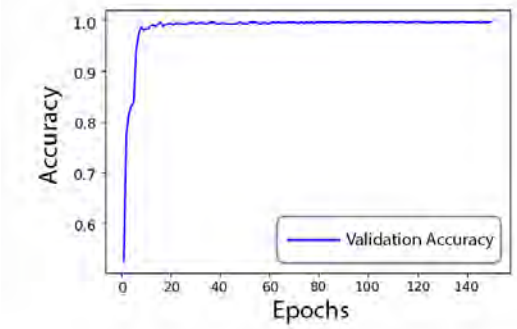
(a) Loss



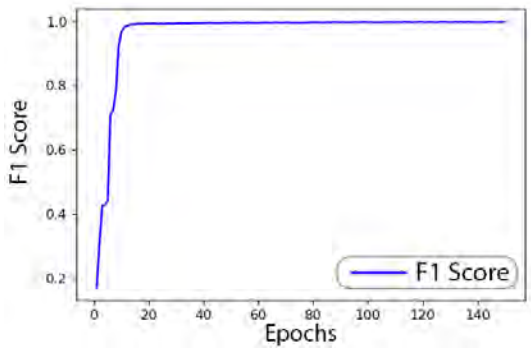
(b) ROC

Figure 3.10: Evaluation Results of Step one (LSTM)

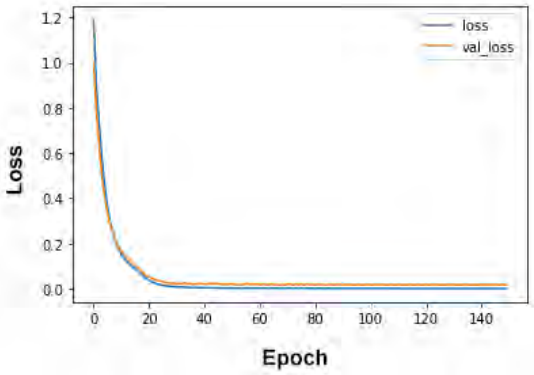
lizing the LSTM-based classifier, leads to significant improvements in both classification accuracy and system efficiency. By introducing the two-step approach of HybridSecNet, we have enhanced the overall accuracy of our intrusion detection system, ensuring more precise and reliable identification of attacks within the CAN network. This comparison underscores the effectiveness of the proposed two-step model in improving the performance of the intrusion detection system.



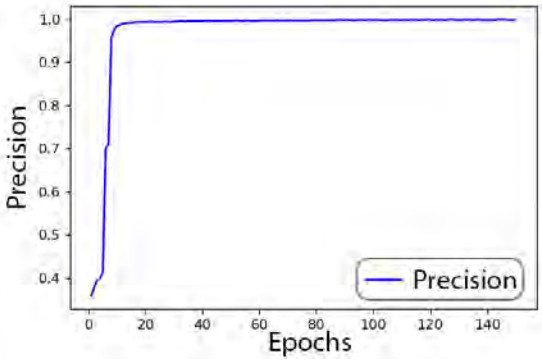
a) Accuracy



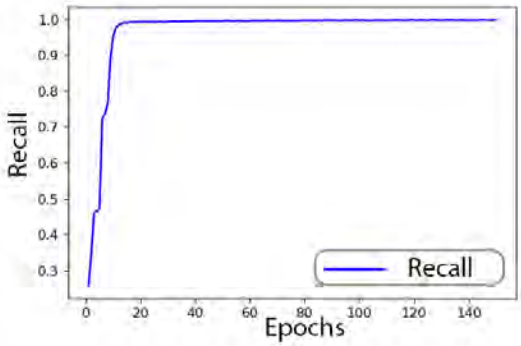
b) F1-Score



c) Loss



d) Precision



e) Recall

Figure 3.11: Evaluation Results of Step Two (LSTM + CNN)

The inclusion of the LSTM-based binary classifier complements the CNN-based multiclass classifier, resulting in a more robust and effective intrusion detection system that outperforms the single-step model in terms of accuracy and efficiency. Table 3.5 compares our proposed two-step model with other multi-step models

Model	Accuracy	Precision	Recall	F1 Score
CNN model	97.84	97.67	97.78	97.86
CNN + LSTM model	**99.579**	**99.583**	**99.579**	**99.580**

Table 3.4: Model Comparison of CNN-based model and CNN with LSTM model

Model	Accuracy	Precision	Recall	F1 Score
NB	83	86	74	76
SVM	90	91	88	89
ANN	92	93	91	92
DT	92	94	78	85
BMFT	89	97	67	78
2-Stage Homogeneous [157]	97.43-99.48	97.44-99.06	97.32-99.21	96.05-99.21
2-Stage Heterogeneous [157]	95.43-98.12	94.19-99.22	95.45-98.23	92.18-98.15
Multi-Stage [158]	99.11	99.13	98.42	99.09
Two-step model (Ours)	**99.579**	**99.583**	**99.579**	**99.580**

Table 3.5: Model Comparison with respect to existing models

that utilize the same dataset, demonstrating the superiority of the proposed Hybrid Two-Step LSTM-CNN Model over existing multi-step models. To further validate the performance of the HybridSecNet model, a comparison is conducted against other two-step models that also use the car hacking dataset, employing accuracy, precision, recall, and F1 score as evaluation metrics. HybridSecNet outperformed both the 2-Stage model proposed by [157] and the Multi-Stage model presented by [158], showcasing its superiority and effectiveness in detecting and classifying attacks.

The focus of the current study is to rigorously assess the prediction time requirements as they relate to the behavior of input samples, as depicted in Figure 3.12. The scenario devoid of vehicular attacks serves as the baseline condition in this analysis. The proposed HybridSecNet model demonstrates consistently higher time efficiency over the CNN-based model across all examined cases in the normal scenario. Notably, as the input sample size increases, the HybridSecNet model maintains reduced prediction times, underscoring the effectiveness of its architectural design. An interesting feature of this model is its ability to bypass the second

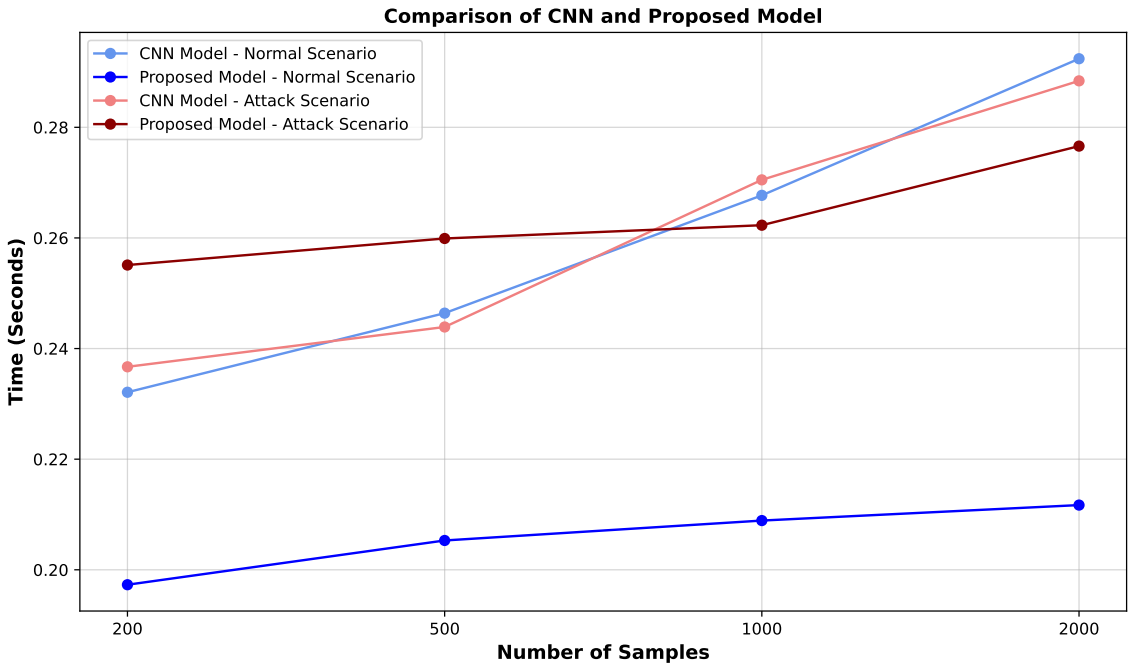


Figure 3.12: Model comparison with respect to time (Seconds) and number of input samples

processing step in the absence of attacks, significantly enhancing its efficiency and processing speed. In contrast, the CNN-based five-class classifier, which is engineered to address both normal conditions and various attack scenarios, requires complete processing of each input sample, leading to prolonged prediction times.

Additionally, the study encompasses scenarios involving vehicular attacks, such as DoS, Fuzzy, Gear, and RPM attacks. In these attack scenarios, which incorporate a mix of normal and attack inputs, the CNN-based classifier initially shows lower prediction times when the sample count is small. However, as the sample volume expands, a critical transition is observed where the HybridSecNet model begins to outperform the CNN model, demonstrating shorter prediction times even in scenarios with attacks. Overall, the findings indicate that the HybridSecNet model excels over the CNN-based classifier with increasing sample counts, suggesting superior performance in real-world conditions where large volumes of inputs are common.

3.4 SCAN-GAN Model

The development of robust security models in the context of applications like VANET and ADAS often faces challenges stemming from limited data availability. The effectiveness of deep learning models relies on having a sufficient volume of data for training and testing. When data is scarce, models may struggle to reach the desired level of performance, or they may lack robustness. To mitigate these issues, data augmentation techniques become essential. Data augmentation involves the generation of additional data points by applying various transformations to the existing dataset. By augmenting the dataset, researchers and practitioners can effectively increase the diversity of the training data, helping the model learn more effectively and become more robust. Data augmentation and synthetic dataset generation techniques play a crucial role in advancing the field of security in VANET and ADAS applications, enabling the development of more effective and resilient models.

3.4.1 Dataset

In this study, we leverage the Car Hacking: Attack & Defense Challenge dataset [71] to facilitate our research. This dataset, curated by the Hacking and Countermeasure Research Lab, Culture Makers, and Korea Internet & Security Agency, was specifically collected from the Hyundai Avante CN7 vehicle model. It encompasses a diverse range of data, comprising both normal and attack messages extracted from the traffic of this vehicle model. The dataset is organized into multiple CSV files, each adhering to a consistent format. The headers include:

- **Timestamp :** Represents the logging time.
- **Arbitration_ID :** Hexadecimal code representing the CAN identifier.
- **DLC :** Data length code.

- **Data :** CAN data field.
- **Class :** Classify each CAN message entry as Normal or Attack. There are 4 types of attacks injected – Replay, Fuzzing, Flooding, and Spoofing.
- **SubClass :** Represents the attack type of each message.

For training, the proposed SCAN-GAN model uses this dataset with spoofing attack data consisting of 12,70,310 CAN messages.

3.4.2 Data pre-processing

The original dataset [71] includes the following features: TimeStamp, Arbitration_ID, DLC, Data, Class, and SubClass. Among these, Arbitration_ID, Data, and Class features were selected as the latent space for model input. Arbitration_ID is represented as three-digit hexadecimal strings. A twofold transformation was conducted to adapt these values for integration into the SCAN-GAN model. Initially, the hexadecimal strings were converted into their corresponding decimal equivalents. Subsequently, these decimal values were normalized to a standardized range between 0 and 1 to ensure compatibility with the model's input requirements.

A similar approach was adopted for other Data fields of the dataset. Each entry in the Data field consists of hexadecimal codes grouped into two digits and separated by spaces. To facilitate further analysis, the spaces were removed, and the resulting hexadecimal code was converted to an integer. The obtained integer values were then normalized to a range between 0 and 1. This preprocessing step was undertaken to enhance the data's suitability for visualization and distribution analysis. The dataset under investigation also encompasses the Class field, which includes two distinct entries: "Normal" and "Attack." To facilitate the subsequent classification task, "Normal" entries were flagged as 0, while "Attack" messages

were assigned a flag of 1. This encoding strategy enabled us to utilize the dataset for training a classifier, with the Arbitration_ID and Data fields serving as features and the Class field as labels, thus providing a basis for comparing the original and synthesized data. It is important to note that within the CAN message dataset, there were instances where the messages were either missing or incomplete. To handle these cases, random values were generated. In particular, for the Arbitration_ID and Data fields, random float values in the range of 0 to 1 were generated, while the Class field was assigned a random value of either 0 or 1.

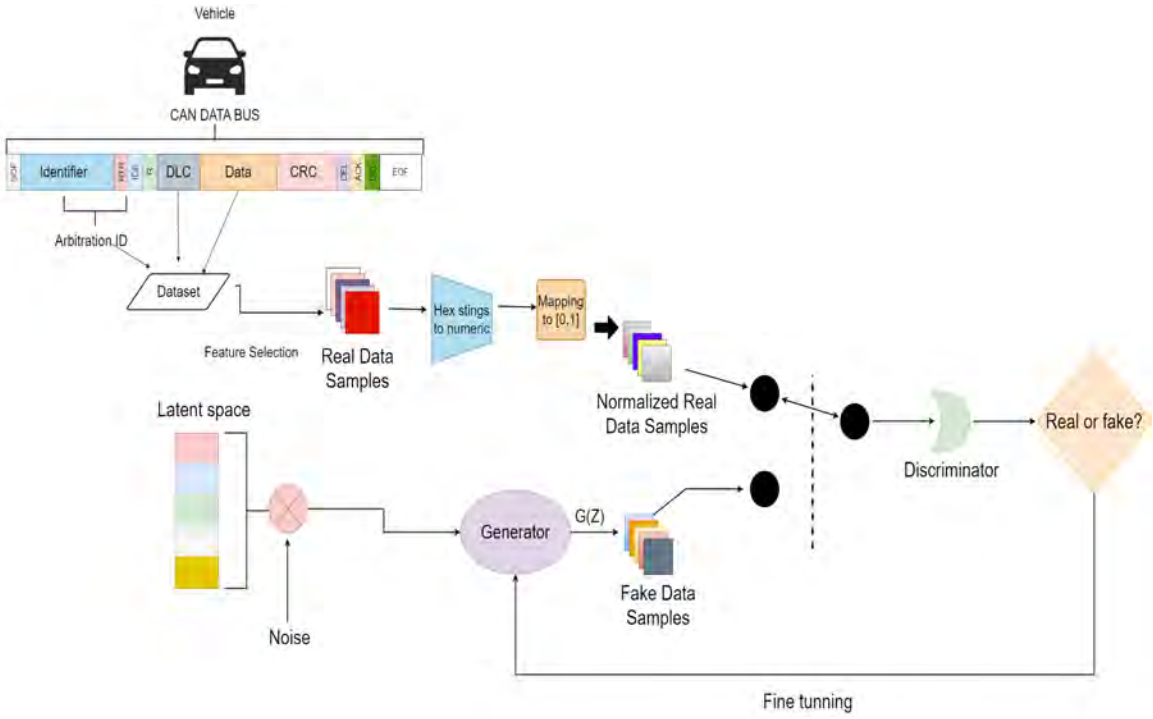


Figure 3.13: GAN architecture

3.4.3 Architecture

SCAN-GAN, short for Synthetic CAN Data Generative Adversarial Network, comprises a generative adversarial network framework with two interconnected networks designed to synthesize data resembling authentic CAN bus data. This GAN architecture harnesses the power of a Generator network and a Discriminator network, which engage in a competitive game. The Generator network is responsible

for producing synthetic data samples, while the Discriminator network evaluates these synthetic samples alongside genuine data samples, discerning whether the input is authentic or fake. This adversarial feedback loop serves as the driving force for training the Generator network, pushing it to generate synthetic data that is increasingly indistinguishable from real data. As a result, SCAN-GAN fosters the creation of synthetic data that closely mirrors the characteristics of genuine CAN bus data. Figure 3.13 illustrates the SCAN-GAN architecture used for generating the CAN dataset. Notably, both the Generator and Discriminator networks exhibit a three-layer structure, reflecting the simplicity and adaptability of this approach, which can be tailored to match the unique characteristics of different datasets.

3.4.3.1 Generator

The Generator network is realized as a sequential model composed of three dense layers. To initiate the synthesis process, latent points are generated from the real dataset and serve as input to the Generator model, which subsequently produces synthetic samples. The first two dense layers within the Generator employ the Rectified Linear Unit (ReLU) activation function, while the final dense layer adopts the Leaky Rectified Linear Unit (Leaky_ReLU) activation function. The use of Leaky_ReLU mitigates the "dead ReLU" issue that can arise when standard ReLU activations consistently maintain values below zero. Additionally, the model's weight initialization employs 'he_uniform' as the kernel initializer, as it has been demonstrated to yield superior performance for layers activated with ReLU. Subsequently, the Generator's loss is determined based on the Discriminator's classification; if the Generator effectively deceives the Discriminator, it receives a reward, but if it fails, a penalty is applied to encourage further improvement in generating synthetic data closely resembling real CAN bus data.

3.4.3.2 Discriminator

The Discriminator network, an essential component of SCAN-GAN, is designed as a three-layer deep sequential model. It plays a pivotal role in distinguishing between the synthetic samples generated by the Generator and the real data points. Similar to the Generator model, the Discriminator incorporates two initial dense layers that employ Rectified Linear Unit (ReLU) activation functions. These layers are responsible for capturing intricate patterns in the input data. However, the final layer of the Discriminator is distinctive; it consists of a single node activated by the 'sigmoid' function. This design choice is made to output a probability, as the Discriminator aims to predict whether the input data is genuine (1) or fake (0). In terms of weight initialization, the 'he_uniform' method is employed as the kernel initializer for the Discriminator model, as it has demonstrated superior performance when dealing with layers activated by ReLU functions. The model is further configured by compiling it with 'accuracy' as the evaluation metric and 'binary_crossentropy' as the loss function. This choice of loss function is appropriate, considering the binary nature of the Discriminator's output, which is either 0 or 1. To optimize the Discriminator's performance, an 'adam' optimizer is utilized.

3.4.3.3 Overall SCAN-GAN model

The SCAN-GAN architecture is formed by the integration of both the Generator and Discriminator models, where the Discriminator's model weights are fixed before initiating the training of the combined model. This approach is implemented to ensure the stability of the adversarial training process. For the compilation of the SCAN-GAN model, the 'adam' optimizer is utilized, which has proven effective in optimizing generative adversarial networks. In addition, the loss function chosen for this combined model is 'binary_crossentropy.' This loss function is well-suited for GANs as it quantifies the divergence between the Discrimina-

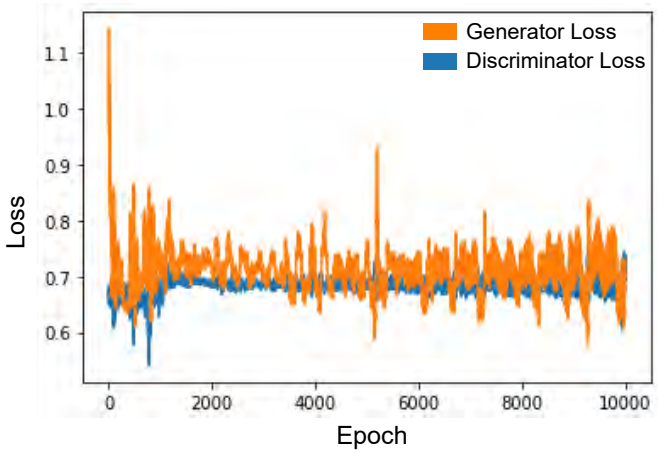
tor's predicted distribution, denoted as $m(y)$, and the actual distribution of the observed data, represented as $n(y)$. The binary cross-entropy loss function quantifies the predictability, or entropy, of $m(y)$ relative to $n(y)$. This metric provides valuable insight into the model's performance by assessing how well it approximates the true distribution of the data labels, $n(y)$.

3.4.4 Training

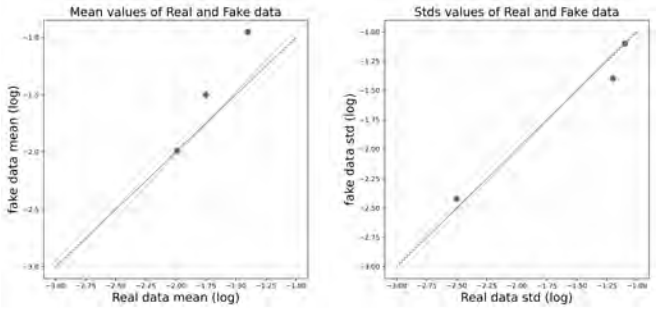
The training regimen for the SCAN-GAN model spans a total of 10,000 epochs, employing a batch size of 128 for each iteration. Within every epoch, a subset of 64 real data samples is randomly selected from the dataset, and an additional 64 data samples are generated by the Generator model. Both sets of samples are subsequently fed into the Discriminator model, which evaluates them and calculates the associated loss values. The overall loss value for each epoch is computed as the average of these two individual loss values. This methodology reflects the ongoing adversarial training process in which the Generator and Discriminator networks continually refine their performance in response to one another. Furthermore, the GAN model is trained using 128 latent points, with the loss values recorded throughout the training process. Figure 3.14a visualizes the progression of these loss values as traverse through the epochs, providing a comprehensive overview of the training dynamics and the convergence of the SCAN-GAN model.

3.4.5 Results

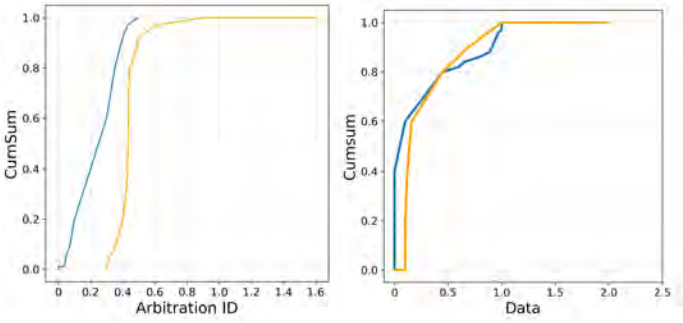
A dataset of substantial scale has been generated, comprising a vast collection of 7.5 million normalized latent points. This dataset encompasses three key features: the Arbitration_ID of the vehicle, the Data field, and the associated label, denoted as "Class," which serves as an indicator of whether a security attack has been detected. Table 3.6 depicts GAN output samples from the generated dataset.



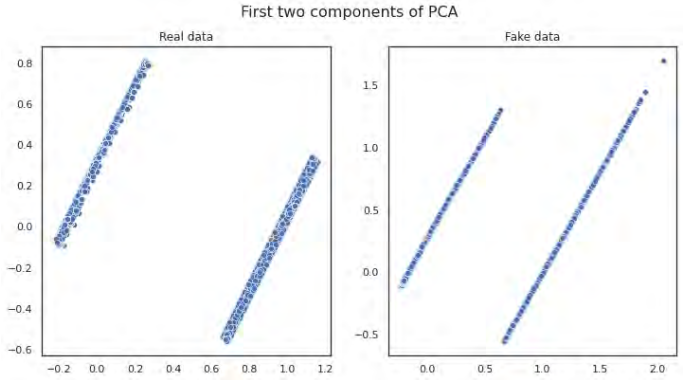
(a) Loss with respect to Epoch



(b) Absolute log mean and std of numeric data of Real & Fake data



(c) Cumulative sums per feature



(d) Principal component analysis (PCA)

Figure 3.14: Evaluation of SCAN-GAN

Sr. No.	Arbitration ID	Data	Class
0	0.3554	0.0561	0
1	0.3743	0.0534	0
2	0.3351	0.1195	0
3	0.5136	0.5254	1
4	0.4068	0.6371	0

Table 3.6: SCAN-GAN generated output samples

The class points undergo a comparison with the mean value of all latent points, leading to their conversion into boolean values. Values less than the mean are denoted as "False," while those greater than the mean are labeled as "True." Subsequently, these boolean values are mapped to binary logic. Additional transformations involve reverting other features to their original hexadecimal codes. New features are introduced, such as the Data Length Code (DLC), computed from the length of the DATA entries in their hexadecimal representation. The timestamp is incorporated based on a predefined starting point and sampled at a specified frequency. Subsequently, the classes are reassigned their string equivalents, where "one" signifies an attack and "zero" represents normal behavior, consistent with the original dataset. Table 3.7 illustrates the data samples from the synthetic dataset following the post-processing steps.

No.	TimeStamp	Arbitration ID	DLC	Data	Class	SubClass
0	10.10164	5FC	8	0D A9 F4 9F 68 7F 5F 00	Normal	Normal
1	10.10224	55C	8	1E 96 30 83 80 7D 94 00	Normal	Normal
2	10.10289	837	8	86 7E 06 15 42 DD 48 00	Attack	Spoofing
3	10.10322	681	8	A3 17 B5 FF 2A ED E8 00	Normal	Normal
4	10.10324	543	8	10 64 B8 3B 70 43 8A 00	Normal	Normal

Table 3.7: Synthetic data samples after post-processing

To assess the fidelity of the generated dataset in emulating the original dataset,

Performance				
Dataset	Logistic Regression	Stochastic Gradient Descent	Multi-layer Perceptron	Quadratic Discriminant Analysis
Original	0.8499	0.8582	0.8582	0.8773
Generated	0.8894	0.8909	0.8748	0.9307

Table 3.8: Performance comparison of the original and generated dataset based on accuracy

both datasets were subjected to a battery of established classifier algorithms. The features used for these classifiers were the "Arbitration_ID" and "Data" columns, while the "class" served as the label for the models. This approach allowed for a comprehensive evaluation of the synthetic dataset's performance in comparison to the original dataset.

The accuracy of the generated dataset was assessed against the original dataset across various classifiers, with consistently superior performance observed in the generated dataset. A summary of these results is presented in Table 3.8, reflecting the superior data quality achieved through the generative process. Table 3.8 demonstrates that the accuracy of several classifiers on the generated dataset surpasses that of the original dataset, underscoring the efficiency of the generated dataset. Furthermore, an alternative methodology was employed to compare and visualize the original and generated datasets. The Python library "table evaluator" was utilized to evaluate the distribution of the real synthetic data. This evaluation was conducted on the normalized versions of both datasets, and the comparison is depicted in Figure 3.14. Additionally, Figure 3.14b provides a comparison of the averages of corresponding normalized data values and the standard deviations of numeric data, represented on a logarithmic scale. Figure 3.14c illustrates the cumulative sums of features in the generated data points compared to the original data. In order to minimize information loss while enhancing data interpretability, Figure 3.14d offers a visual representation of principal component analysis for real and synthetic data points, facilitating a multidimensional data visualization

for comparative purposes.

3.5 Summary

This chapter introduced three innovative approaches that hold great promise in the realm of security for ADAS-enabled vehicles. Firstly, the MbRE IDS, a lightweight deep learning-based intrusion detection system, stands out for its ability to achieve outstanding detection performance across 19 distinct vehicular behaviors tailored for VANET edge scenarios. Secondly, the HybridSecNet, a two-step model combining LSTM and CNN techniques, excels in the realm of intrusion detection within in-vehicle communication systems, particularly in CAN. This model not only achieves remarkable accuracy but also sets the stage for elevating in-vehicle security beyond the capabilities of single-step intrusion detection systems. Lastly, the SCAN-GAN, a Generative Adversarial Network (GAN)-based method, offers an efficient approach for synthesizing VANET datasets. It not only provides precise data generation but also holds potential for customization and the creation of data with specific features and patterns. These approaches collectively contribute to the advancement of security in ADAS-enabled vehicles, addressing the critical need for robust security measures in modern vehicular networks.

Chapter 4

Enhancing Driver Attention Behind the Wheel: Recognizing Short-Term Microsleep Using EEG in ADAS

4.1 Introduction

The evolution of personal cars has been intrinsically linked to economic progress and the financial well-being of individuals. As the economy has improved, people have increasingly favoured personal vehicles for their convenience, comfort, and efficient time management over public transportation. This shift in preference has led to a surge in the use of personal cars. Simultaneously, the automotive industry has undergone a significant transformation, progressing from early steam engines to innovations in petrol and diesel-based vehicles and finally to the latest advancements in electric cars. This progression has not only been about fuel types but also significant improvements in vehicle features, culminating in the development of ADAS, which have become a ubiquitous presence on today's roads. Unlike fully autonomous vehicles, which offer complete automation, ADAS-enabled ve-

hicles fall into the category of semi-autonomous vehicles. These vehicles rely on a driver's presence behind the steering wheel for several critical functionalities. These functionalities include Adaptive Cruise Control (ACC), where the vehicle automatically adjusts its speed to maintain a safe distance from surrounding cars and objects. However, the driver is required to remain alert, steer, and apply the brakes when specific dangerous situations arise. Another crucial functionality is lane-keeping assistance, which gently guides the vehicle within its lane. The driver is expected to hold the steering wheel and remain engaged, as the system may disengage if it detects the driver's lack of involvement. On highways and at intersections connecting to main roads, there is a risk that ADAS-based vehicles might not execute proper lane changes or acceleration and could fail to identify oncoming vehicles, potentially resulting in collisions. In such scenarios, the driver's presence and active involvement become paramount. The driver needs to keep their hands on the steering wheel, their eyes on the road, and their attention on both the surrounding traffic and their own vehicle's behaviour. They should be ready to use manual override to take control in case the system's performance is compromised. In summary, despite the sophisticated technology integrated into ADAS-enabled vehicles, the human element remains critical in ensuring safe, responsible, and adaptive driving.

Nevertheless, even with the advanced assistance provided by ADAS-enabled vehicles, a significant number of accidents are still attributed to various human factors. Driver-related issues, such as inattentiveness, drowsiness, fatigue, or incomplete sleep, account for a considerable percentage of road incidents. Among these, microsleep episodes represent a complex and major contributing factor to accidents, posing a substantial challenge for detection and mitigation. These episodes involve brief, unintended lapses in attention, lasting only a few seconds, yet they can have catastrophic consequences, endangering the driver, passengers, and other road users. Microsleep, characterized by momentary, uncontrollable bouts

of sleep during wakefulness, can lead to a temporary loss of awareness, rendering the driver oblivious to their surroundings and incapable of responding to sudden or unexpected events. This issue underscores the urgent need for an effective solution to detect, manage, and ultimately prevent microsleep episodes among drivers. This chapter is dedicated to addressing the critical issue of microsleep detection among drivers in ADAS, ultimately enhancing a driver's attentiveness behind the steering wheel. The proposed solution leverages Electroencephalography (EEG), a technology designed to record and monitor electrical activity in the brain. EEG data is meticulously pre-processed using Wavelet transform and Short Time Fourier Transform (STFT) techniques before being fed into our developed deep learning-based model for microsleep detection. In addition, the hardware implementation demonstrates the model's real-world application and use cases, effectively detecting the probability of wakefulness and microsleep states in real-time while providing alert mechanisms to ensure driver safety. This research marks a significant stride towards addressing a pressing concern in road safety and driver assistance systems. The key contributions of this chapter are as follows:

1. In our research, we introduce a novel deep learning model termed the "Diamond Model." This model is designed to accurately assess the sleep stage of raw single-channel EEG data, presenting a significant advancement in sleep stage classification.

2. In our study, we showcase an attention-based approach that combines the Wavelet transform with an epoch's Short Time Fourier Transform (STFT) Spectrogram to extract more comprehensive and richer features.

3. A sophisticated data conversion and cleaning pipeline was deployed with a weighted cross-entropy loss function to deal with the class imbalance in the dataset.

4. Furthermore, we introduce an innovative pipeline that encompasses the

hardware implementation of a real-time microsleep detection model. This pioneering approach enables the practical application of our model for on-road microsleep detection, contributing to improved driver safety and attention monitoring in real-world scenarios.

The subsequent sections of this chapter are structured as follows: Section 4.2 introduces the Dataset Selection Methodology, delineating the criteria and process used for dataset selection. Section 4.3 provides a comprehensive overview of the data pre-processing steps carried out to prepare the dataset for further analysis. Section 4.4 presents the proposed WT-STFT algorithm, detailing its methodology and implementation. This algorithm is a key component of microsleep detection. Subsequently, Section 4.5 explores the deep learning aspects of the proposed model. This section discusses the architectural design and functionality of the "Microsleep Diamond Model," which is a crucial element in the microsleep detection framework. Section 4.6 addresses the hardware implementation of the proposed work, focusing on the practical application and deployment of the microsleep detection system. Section 4.7, delivers and analyzes the results obtained from the proposed microsleep detection framework, providing insights into its performance and efficacy. Finally, Section 4.8 serves as a summary, encapsulating the key findings, contributions, and implications of the work presented in this chapter.

4.2 Dataset Selection Methodology

The concept of Microsleep Episodes (MSE) is central to this classification, and it relies on a combination of three key indicators: (1) neurophysiological parameters, (2) psychomotor performance measures, and (3) eye and/or facial behaviour. Consequently, it has long been the standard practice to record both EEG and video data when analyzing sleep stages using the Maintenance of Wakefulness

Test (MWT). As depicted in Figure 4.1, the raw data samples illustrate instances of deep sleep and wakefulness states within the MWT dataset.

The MWT test assesses an individual's ability to remain awake while resisting the urge to fall asleep in sleep-inducing situations, a process commonly referred to as "wake tendency." Among the various tests administered in clinical settings, the MWT closely simulates passive real-life situations, making it especially relevant for the study of MSEs. Moreover, the latency of sleep identification in the MWT dataset exhibits a strong correlation with real-world driving performance, further emphasizing its significance. This enables precise and high-resolution classification of MSEs within the MWT dataset, aligning with the conditions resembling how a driver sits in a car during low-light driving scenarios. Throughout previous research on microsleep detection, these MWT tests have frequently served as standard benchmarks and parameters for microsleep detection tasks. In the present study, we utilize the Maintenance of Wakefulness Test (MWT) dataset [2, 3], which comprises data from 64 subjects aged between 30 and 70 years. Each file within

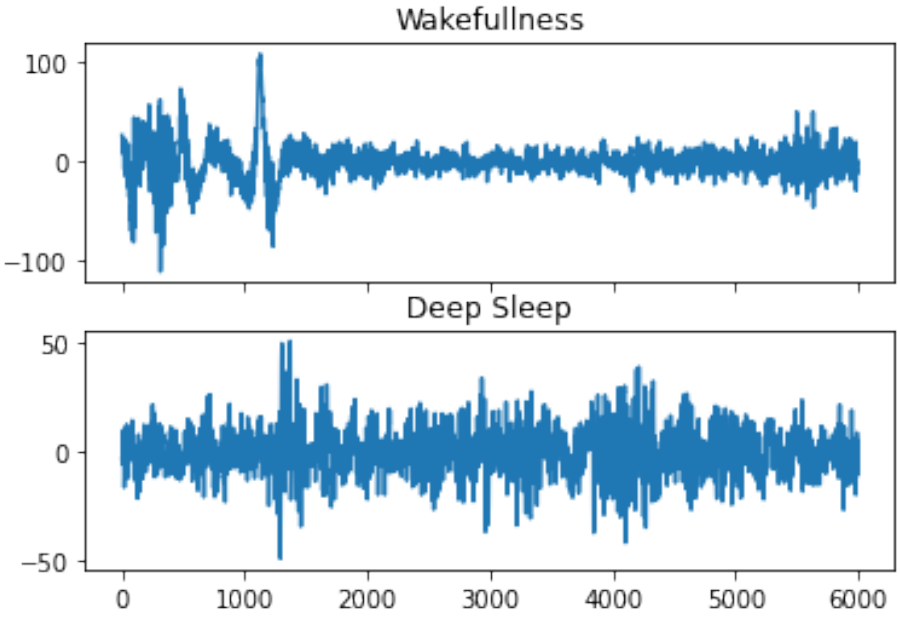


Figure 4.1: Raw Data Samples of Deep Sleep and Wakefulness state of the same subject from the MWT dataset [2, 3].

Age Group (Year Range)	Female (Count)	Male (Count)	Total (Count)
31-40	10	8	18
41-50	10	10	20
51-60	10	5	15
61-70	7	4	11
Total (Count)	37	27	64

Table 4.1: MWT Dataset Demographics [3].

the dataset contains a recording of an MWT trial conducted in the early afternoon. The testing environment includes bedrooms insulated from external light and equipped with dim lighting, mirroring conditions akin to low-light driving scenarios. The dataset provides labels for wakefulness, episodes of drowsiness, microsleep episodes, and microsleep episode candidates. For our analysis, we focus on identifying "MSE" and "Wakefulness" labels, while the other two denote drowsiness. The demographic distribution of the test subjects, as summarized in Table 4.1, sheds light on the population demographics typically susceptible to microsleep episodes while driving.

4.3 Data Pre-processing

Addressing the inherent noise and frequency defects within the input data is pivotal in the presented pipeline's design philosophy. The primary objective is to execute fundamental filtering and modulation processes, thereby standardizing the data into a format that proves useful for a variety of applications while retaining the essential signal characteristics. This entails the segmentation of the continuous raw single-channel EEG into a series of 30-second epochs, facilitating the extraction of EEG epochs from the provided raw signal. Subsequently, we employ a comprehensive pre-processing pipeline to manually extract valuable information from these epochs before forwarding the data to the neural network for further analysis. Figure 4.2 illustrates the proposed pre-processing approach, and the following subsections delve into the specifics of each individual stage.

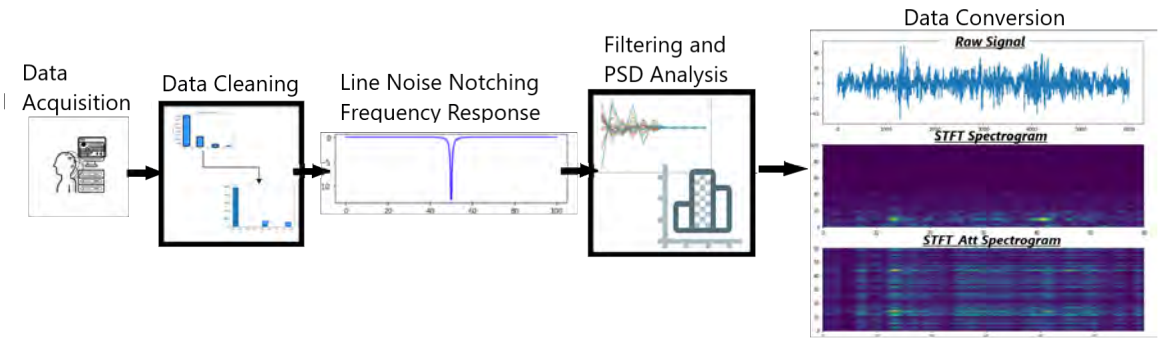


Figure 4.2: Pre-Processing Pipeline for the model

4.3.1 Line Noise Removal

In the initial preprocessing of the original dataset, a band-pass filter with a range of 0.5-75 Hz was applied, aiming to capture the relevant signal frequencies. To further enhance the data quality, a 50 Hz second-order infinite impulse response (IIR) notch filter was employed to eliminate noise originating from the power grid's mains frequency. Notably, this high-quality notch filter was utilized to precisely target and suppress the interference within a narrow frequency band, while preserving the integrity of the broader spectrum. In alignment with the work by Nima Bigdely-Shamlo *et al.* [159], it was recognized that long-term signal non-stationarities can affect the output of the notch filter. Therefore, to mitigate this, a 1 Hz high-pass filtering step was introduced before applying the notch filter. This high-pass filtering was executed using a 4th-order Butterworth (sos) filter for optimal noise removal and signal preservation.

4.3.2 Filtering

To extract the essential signal components within the frequency range of 4-14 Hz, an IIR filter was thoughtfully employed. While an ideal FIR filter could have been a viable alternative, its computational demands and susceptibility to noise artifacts were taken into consideration [160]. Consequently, a biquad Chebyshev II filter was chosen as the bandpass filter for its efficiency and robustness. Following

this filtering stage, power spectrum density analysis was employed to eliminate unwanted artifacts from the signal. The designed filter's characteristics and response can be observed in Figure 4.3. This comprehensive approach ensures the extraction of relevant signal components while mitigating the impact of computational overhead and susceptibility to noise artifacts.

4.3.3 Data Conversion

The pivotal step in our pre-processing pipeline involved transforming raw signals into a usable format. We opted for a two-pronged ensemble approach, with further flexibility in the tuning of hyper-parameters like window size, taper function, and precision. The temporal data conversion method focused on generating datasets that retained signals in their original time-amplitude form, making use of direct raw data inputs. This method offers the advantage of aligning with human scoring in visual pattern recognition tasks, enabling a deeper understanding of signal fluctuations, particularly at boundaries that are often tapered in spectral computations. Additionally, we created a separate dataset of linearly normalized data, which facilitates faster convergence rates during training. This resulted in

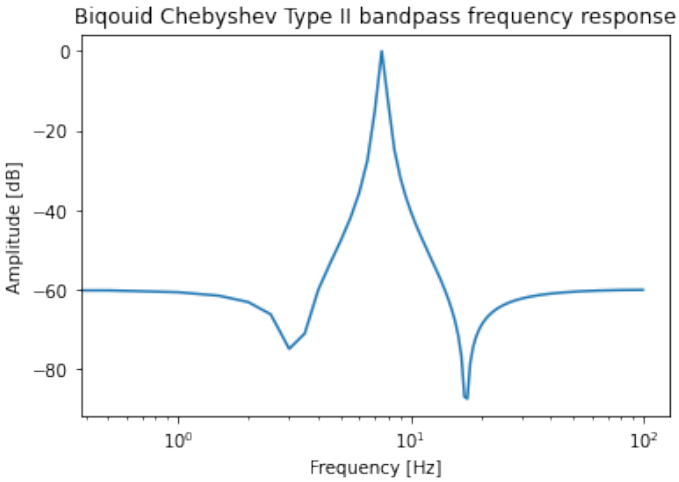


Figure 4.3: Ideal frequency response of our designed Biqouid Chebyshev Type II band-pass filter

two distinct datasets: *RAW*, representing the original pre-processed signal list, and *NORM*, signifying the normalized version of the same signal.

The spectral analysis aspect incorporated Short-Time Fourier Transform (STFT) and the Wavelet Transform-STFT fusion modes, detailed in Section 4.4. These datasets are termed STFT and STFT_Att, respectively. While the full Fast Fourier Transform (FFT) is known to be sensitive to noise and non-stationarities, it offers high spectral precision, which can be addressed using Welch's transforms. Additionally, the Wavelet Transform method builds the time-frequency plane one frequency point at a time, employing a sliding window approach. Our model is designed to harness both spectral and temporal dynamics while leveraging the individual strengths of the aforementioned methods. Thus, we propose an attention map constructed by combining a custom Short-Time Welch's paradigm with a Wavelet-filtered product. This map is subsequently convoluted over an STFT to yield a final noise-free output, advancing the quality and utility of the pre-processed data.

4.3.4 Class Specific Loss Gradients

Our model uses a multi-class cross-entropy loss function and is defined as the average loss value for a training batch:

$$\hat{Loss} = -1/K * \sum_{1}^{K} \sum_{i=1}^{outputsize} y_i * \log \hat{y}_i \tag{4.1}$$

Where $\hat{y}_i$ is the model output's i-th scalar value, y_i is the ground truth, and K is the total number of samples. This loss function we employ is proficient at distinguishing between two discrete probability distributions. To define this function, we use y_i to denote the likelihood of event i occurring. It is crucial to emphasize that the sum of all y_i values equals one, indicating that only one event can occur at a time. The negative loss gradient comes into play to penalize devia-

tions in cases where the probability distribution becomes closer to one another. This property of the loss function contributes to its effectiveness in discerning and optimizing for distinct events. The MWT dataset exhibits a substantial class imbalance, characterized by a pronounced bias towards the "Wakefulness" class. Even after consolidating class labels "2" and "3" into a single label, the issue of class imbalance persisted. Furthermore, the previously mentioned loss function primarily focuses on distinguishing between any two given labels, leading to uniform penalization of incorrect classifications across all classes. Consequently, the trained model tends to develop a biased preference for the majority class. To mitigate this imbalance and ensure fair learning, we adopted a weighted cross-entropy loss. This modified version of the loss function effectively addresses the dataset's imbalance by assigning weights to different classes.

$$W_c = H_c * (K_c/K) \tag{4.2}$$

$$\hat{Loss} = -W_c/K * \sum_{1}^{K} \sum_{i=1}^{outputsize} y_i * \log \hat{y}_i \tag{4.3}$$

where W_c represents the weight assigned to that class , H_c is a tunable hyperparameter, and K_c/K is the ratio of number of samples in class c. Thus, by compounding a W_c we aim to nullify the effects of class imbalance.

4.4 WT-STFT Algorithm

In our approach, we leverage the Wavelet Transform, which offers distinct advantages over traditional methods such as the Fourier Transform. The choice of the Wavelet Transform is motivated by its ability to effectively capture the inherent dynamics of EEG signals, which exhibit temporal variability. Time and frequency are pivotal parameters when studying EEG signals as they provide critical insights into the signal's characteristics. Every peak, dip, or subtle change within

the signal plays a significant role in pattern recognition for applications related to EEG signals. In scenarios where abrupt changes are of paramount importance, Fourier Transform falls short in its ability to provide meaningful representations. This limitation is addressed by the Wavelet Transform. The Wavelet Transform excels in scenarios where signals evolve over time, making it a preferred choice. Unlike the Fourier Transform, which predominantly provides information about a signal's frequency content, the Wavelet Transform empowers us to simultaneously explore both time and frequency domains. This capability is instrumental when dealing with dynamic signals. The adaptability and customizability of the wavelet transform further set it apart. Wavelets can be tailored to slide with the signal, and their size and position can be adjusted as needed. This flexibility enables precise control over signal processing and analysis, making it possible to apply wavelets to various sections of the signal. Furthermore, the Wavelet Transform offers an essential feature, localization. It allows us to precisely determine when and where significant events or changes occur within the data, rendering it invaluable for feature extraction. Additionally, the concept of Multiresolution Analysis becomes achievable with the Wavelet Transform. Larger wavelets can be employed to capture slower changes in the signal, while smaller ones can be used to detect rapid variations within the same signal. This versatility and adaptability make the Wavelet Transform a powerful tool for analyzing EEG signals with a focus on both temporal and frequency aspects, which is of paramount importance in EEG-based applications. Our proposed model leverages an attention-based mechanism that combines the advantages offered by the Wavelet transform with the Short Time Fourier Transform (STFT) Spectrogram.

4.4.1 Wavelet Transform

The Fourier Transform is highly effective at capturing the global frequency information of a signal. However, in many applications where we need to analyze

signals with shorter time periods, such as electrocardiography (ECG), this type of signal analysis may not be suitable. The Wavelet Transform, as introduced by Daubechies in 1990, offers an excellent alternative. The Wavelet Transform (WT) decomposes a function into a set of wavelets, making it particularly well-suited for applications where the temporal characteristics of signal variations are of paramount importance. Wavelet transforms come with several advantages. They can simultaneously extract local spectral and temporal information from the input signal, offering a more comprehensive view of the data. Additionally, they provide flexibility in choosing from a range of wavelets, such as Morlet, Haar, and Daubechies, to best suit the specific analysis needs. The WT divides the input signal into time windows of a specific size and processes each time window individually, akin to the Short-term Fourier Transform (STFT). By using a broader time window, the WT can enhance the frequency resolution of the output signal, although at the cost of reduced time resolution. The Wavelet transform produces individual coefficients of a set of orthonormal functions (wavelets), essentially breaking down the incoming signal into its wavelet components. Due to its significant advantages, such as better localization of time-frequency characteristics, the continuous Wavelet Transform has gained substantial popularity across various domains, as discussed in Merry's work in 2005 [161]. This preference for the continuous Wavelet Transform has often surpassed its discrete counterpart. A wavelet transform of a signal $X(t)$ is defined as follows:

$$T(a,b) = \frac{1}{\sqrt{b}} \int_{-\infty}^{+\infty} X(t).\theta(\frac{t-a}{b})\, dt \tag{4.4}$$

where θ is a wavelet function, and b and a are scaling and translating factors, respectively.

4.4.2 Proposed Algorithm

Our model is designed to harness both spectral and temporal dynamics while taking into consideration the unique advantages offered by the methods, as discussed in the previous section. As part of our approach, we propose the use of an attention map created by combining a custom Short-Time-Welch paradigm with a Wavelet-filtered product. This attention map is subsequently convolved over a Short Time Fourier Transform (STFT) to obtain a final, noise-free output. To extract the normal spectro-temporal image using the Wavelet Transform (WT), we decompose a 30-second EEG epoch (sampled at 200Hz frequency) into one-second frames with a 50% overlap. Furthermore, the transformation of the original spectrogram to the frequency domain is accomplished using a 200-point STFT, aided by Hamming Window convolution. Our final Wavelet Transform representation has time on the x-axis, with a maximum of 29 time frames, and the y-axis consists of frequency bins. We carefully calibrate the frequency bin count to ensure it aligns with multiples of the convolutional unit's shape, setting it to 128. These representations are then normalized prior to training and testing, ensuring consistency and effective model performance. Additionally, it explores the intriguing relationship between Short-Time Fourier Transform (STFT) and Wavelet Transform (WT) in the context of spectro-temporal image generation. Both these methods, quantitatively, yield similar spectro-temporal representations, depicting time on the x-axis and frequency on the y-axis. However, they differ fundamentally in their algorithmic approaches. The STFT involves segmenting the time signal into shorter epochs and computing the Fourier transform on each of these segments independently. Conversely, the WT progressively works with a narrower set of frequencies while processing the entire time domain.

Drawing inspiration from the principles of self-attention units, particularly as outlined in [162], the proposed technique introduce a novel approach. We modify the

traditional notion of covariance matrices by taking the dot product of the STFT and the transpose of the WT spectrograms, formally defined as $S.\dot{W}$, where S represents the STFT with 100 frequency bins and W the WT with 100 width length. This dot product operation is performed along the time axis to fuse the information from both transforms. To ensure compatibility with subsequent neural network processing the resulting fused image is normalized. In Figure 4.2, the final column showcases the progression from the raw signal to the STFT spectrogram and, finally, to the STFT-WT fused spectrogram. Given its derivation from the principles of attention mechanisms, we denote this data conversion as "STFT_Att" for the purpose of discussion within this paper.

4.5 Diamond Model

The transmission of EEG signals through a network can result in a loss of signal clarity, which in turn affects the representation of deep characteristics. To address these challenges and enable the identification of both intra and inter-features of EEG epochs for predicting microsleep, we introduce the proposed Diamond model. In the following two sub-sections, we provide a comprehensive overview of the construction of the Diamond model, delving into the model's architecture and design details.

4.5.1 Part 1 - Time independent section

Part 1 of the model architecture, which focuses on time-independent features, is represented in Figure 4.4. The Main Unit (MU) of the model is organized into three key areas: a Convolutional layer (Conv1D for temporal input and Conv2D for spectrogram input), followed by a Rectified Linear Unit (ReLU) non-linearity, and finally, a Batch Normalization layer. Two primary branches within the Main Unit (MU) have distinct filter sizes - one with a small filter size and another with

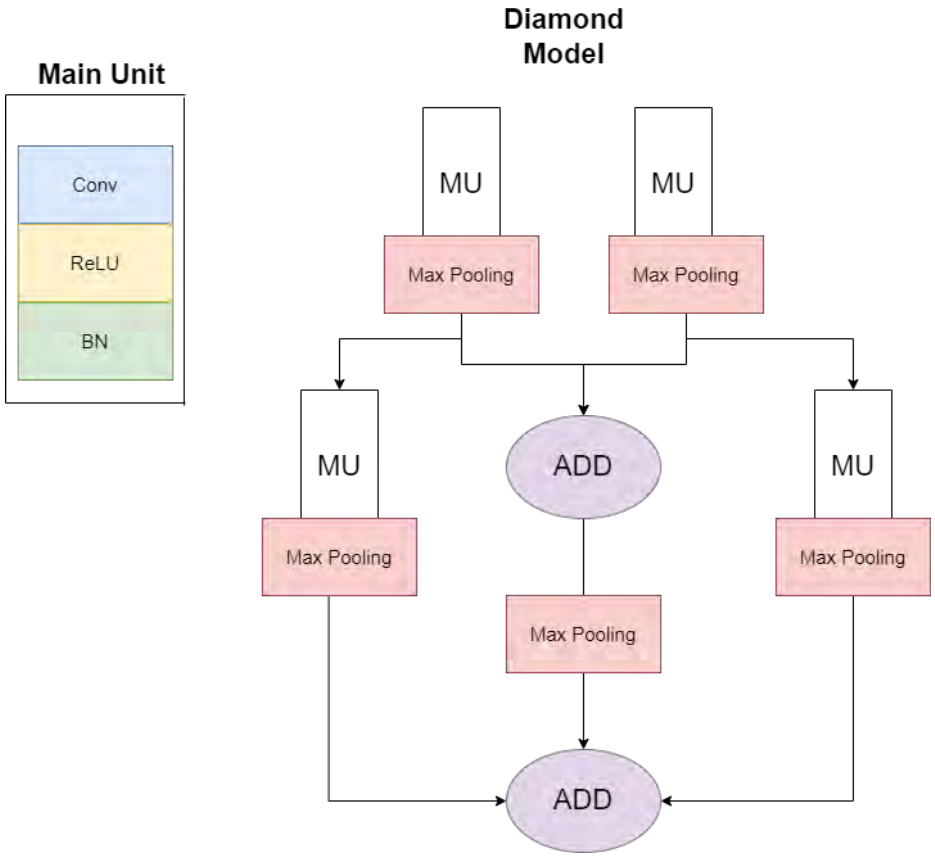


Figure 4.4: Diamond Model : Time Independent Part of the model

a larger filter size. This design choice allows us to extract both temporal information (related to the appearance of specific EEG patterns) and spectral information. The concept of variable filter sizes was inspired by signal processing techniques, offering a trade-off between capturing time-domain and frequency-domain data [163]. Therefore, we leverage both time-domain and spectral features to enhance the classification task. Each block within these branches is complemented by a pooling layer that downsamples the inputs through max pooling operations. Throughout this process, the convolutional filter size was consistently set at 64, and padding was configured as "same." Notably, the kernel size of one branch was set to be half of the other, with kernel sizes of (3,6) and (6,12) for the left and right branches, respectively. To further enhance the model's capabilities in preserving temporal resolution and facilitating the learning process, an additional shortcut residual connection was incorporated. This reformulation of the computation em-

ploys an attention-type mechanism, which proves advantageous in addressing issues like the dying ReLU problem. The entire model can be concisely represented through the following set of equations,

$$y_1 = MP(MU_6(MP(MU_3(x)))) \tag{4.5}$$

$$y_2 = MP(MU_{12}(MP(MU_6(x)))) \tag{4.6}$$

$$y_3 = MP(MP(MU_3(x)) + MP(MU_6(X))) \tag{4.7}$$

$$Y = y_1 + y_2 + y_3 \tag{4.8}$$

where, y_1, y_2 and y_3 went through element-wise addition and the numbers in the subscripts represents the kernel sizes. Y is the final output of this section of the model. Further MU_x represent Main Unit with a kernel size of $(x, 2x)$.

4.5.2 Part 2 - Time dependent section

In the second stage of the model (Figure 4.5), the incoming data undergoes a max pooling layer to downsample its representations. To learn temporal dependencies, particularly stage transition rules within the epoch, we employ bidirectional recurrent neural network (BiRNN) [164] units instead of traditional LSTM units. Bidirectional LSTMs offer the advantage of being able to memorize whether an epoch transitioning from wakefulness to microsleep or vice versa. BiRNNs possess the capability to process data in both forward and backward directions, overcoming the unidirectional nature of standard RNNs. Consequently, the current state within the BiRNN has simultaneous access to both preceding and subsequent input data. The input data stream is fed into the forward network in its normal time order and into the backward network in reverse time order. The final output of the BiRNN is defined as a linear combination of the outputs from these

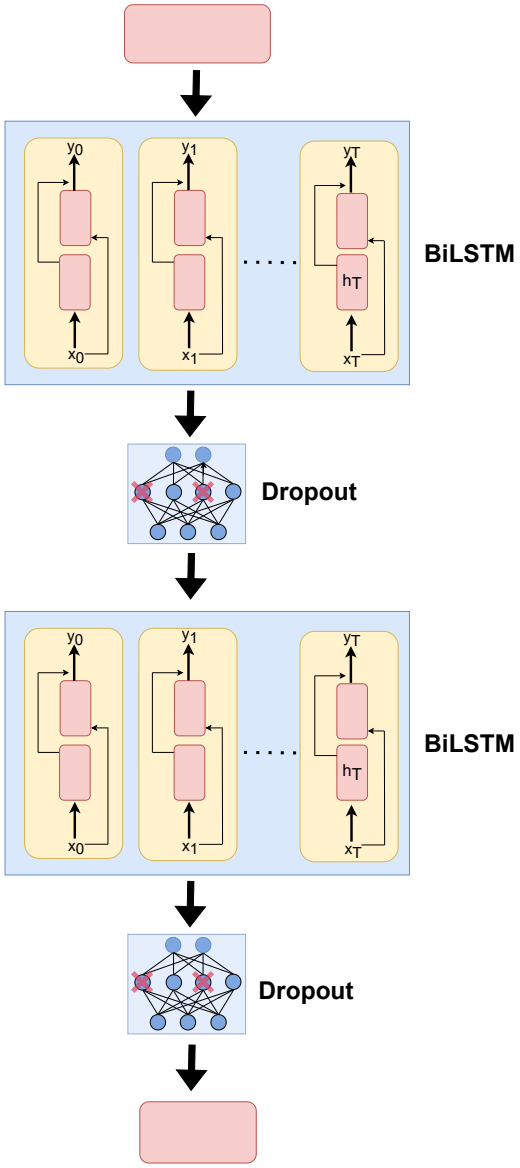


Figure 4.5: Diamond Model: Time-Dependent section of the model

two networks.

$$h_t^f, c_t^f = \tanh\left(W_{xt} + V_{h(t-1)} + b\right) \tag{4.9}$$

$$h_t^b, c_t^b = \tanh\left(W_{xt} + V_{h(t+1)} + b\right) \tag{4.10}$$

$$y_t = h_t^b || h_t^f \tag{4.11}$$

where h_t^f and b are the hidden state and the bias of the forward network. Also, xt and yt are the input sequence and the concatenated output vector of the model,

respectively. During both the training and testing phases, a crucial practice we implement is the re-initialization of cell states in the equations mentioned above. This re-initialization is performed at the start of each patient data set, ensuring that the model exclusively relies on temporal data from the current subject. In our Long Short-Term Memory (LSTM) units, we incorporate peephole connections, a mechanism that enables the gating components to consider the current memory state before making updates. Following each Bidirectional LSTM (BiLSTM) unit, we introduce a Rectified Linear Unit (ReLU) non-linearity. Subsequently, we apply a dropout rate of 50%, randomly setting input values to 0 during the training phase. This dropout procedure serves to enhance the model's robustness and ability to generalize. To generate final predictions, a softmax-based Dense layer is employed, enabling the model to output class probabilities for microsleep prediction.

4.6 Hardware Implementation

The hardware implementation of the framework presented above is carried out using the NeuroSky MindWave Mobile EEG device [165] to detect the occurrence of microsleep in real-time. Drivers are required to wear this EEG device, as illustrated in Figure 4.6, during their journeys. Once worn, the Signal Acquisition process operates seamlessly in the background, continuously collecting signals from the headset and transmitting them to the Jetson Nano via Bluetooth using the PyBluez module, an open-source Python module tailored for Bluetooth device interfacing [166]. The Diamond model, integrated with the Jetson Nano, exhibits the capability to discern incoming EEG data into distinct sleep stages, such as Deep sleep, Microsleep, or wakefulness. Furthermore, the Streamlit user interface (UI) offers drivers a means to monitor their sleep stages and ensures that the alert system, as described in algorithms 3 and 4, is in operation. The holistic hardware

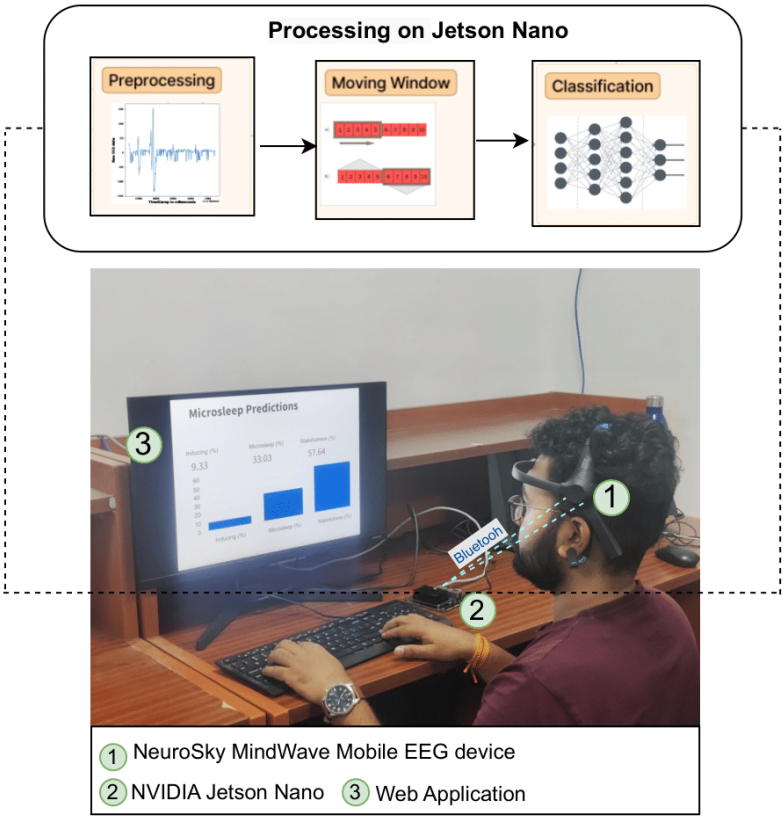


Figure 4.6: Proposed pipeline for microsleep detection

implementation pipeline, as outlined in Algorithm 2, serves as the pseudocode governing the EEG signal acquisition and subsequent Bluetooth transmission via the Jetson Nano. This process encapsulates the signals and subjects them to a sleep stage classification procedure utilizing diamond model [167] and EEG headset.

4.7 Results

The data pre-processing pipeline yields four distinct types of input vectors for our deep learning model: RAW, NORM, STFT, and STFT_Att. The data is divided into training and test sets with an 80:20 ratio, and the training data is further divided into a 70:30 ratio for training and validation, respectively. To ensure a uniform distribution of all classes, the data is stratified, and class weights are appropriately adjusted. For model training, we explore two values of an initial learning rate and

Algorithm 2 Pseudo-Code for “EEG Signal Acquisition, Processing, and Sleep Stage Classification"

```
Initialize EEG_Headset
Initialize Bluetooth_Connection
Bluetooth_Socket = Bluetooth_Connection.Create_Socket()
Bluetooth_Connection.Pair (EEG_Headset)
Bluetooth_Connection.Authenticate (EEG_Headset)
while True do
    EEG_Signal = EEG_Headset.Read_Signal()
    Processed_Signal = Preprocess (EEG_Signal)
    Features = Extract_Features (Processed_Signal)
    Sleep_Stage = Sleep_Stage_Model.Classify (Features)
    Output (Sleep_Stage)
    if Sleep_Stage == “microsleep" then
        Alert_Driver()
    end if
end while
Bluetooth_Connection.Close_Socket (Bluetooth_Socket)
```

Algorithm 3 Psuedo-Code for "Position based multi path flooding algorithm to send alert"

Echo Distance $\leftarrow D_e$
Threshold Distance $\leftarrow D_t$
New neighbor vehicle ID $\leftarrow V_n$
Old neighbor vehicle ID $\leftarrow V_o$

Create cluster - Send echo packet from RSU to all cars within range.

if $D_e \leq D_t$ **then**
    Add vehicle ID into cluster
end if

At Source - Send alert message to nearest RSU

At RSU - Send alert message within cluster to all vehicles.

if $V_n = V_o$ **then**
    Drop transmission to avoid Repetition
end if

At Destination - For each message from RSU, Send an acknowledgement packet.

Algorithm 4 Psuedo-Code for "Alert system for pedestrian's mobile device"

Echo Distance $\leftarrow D_{echo}$
Threshold Distance $\leftarrow D_{thresh}$
new pedestrian's mobile device ID $\leftarrow P_{Newid}$
Old pedestrian's mobile device ID $\leftarrow P_{Oldid}$

Create cluster - Send echo packet from RSU to pedestrian's mobile device within range.

if $D_{echo} \leq D_{thresh}$ **then**
    Add the pedestrian's mobile device ID into a cluster
end if

At RSU - Send alert messages within a cluster to all pedestrians' mobile devices.

if $P_{Newid} = P_{Oldid}$ **then**
    Drop transmission to avoid Repetition
end if

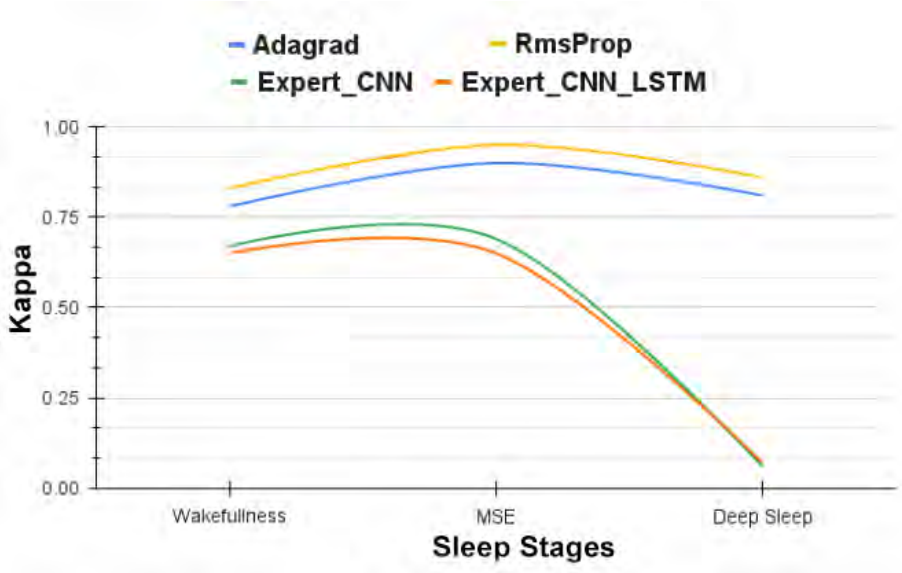


Figure 4.7: Individual Kappa Values of the RAW model with different parameters from the test dataset, comparing it with results from [4].

two different optimizers in a recursive combination to determine the optimal set of parameters. Figure 4.7 provides an explicit visualization of the statistical variation in these models across the three output classes.

Based on the Kappa values, it is evident that among the four models, the one utilizing the RMSprop optimizer stands out as the quantitative best performer.

On the other hand, the model employing the Adagrad optimizer is the qualitative best performer, mainly due to its consistent and flat response.

Figure 4.8 represents a comprehensive overview of the results, presenting different bar graphs for sets of testing matrices across all the combinations of models that were trained. Notably, the RAW_1 and STFT_Att_2 methods stand out with outstanding results, underscoring the practicality of the RAW unit and the efficacy of the signal processing approaches we've proposed. The results show that all the matrices perform optimally with the temporal mode, yielding an overall increase of approximately 1% in testing accuracy when compared to spectral analysis. Within spectral analysis, the use of STFT_Att leads to an improvement of approximately 0.4% in testing accuracy and a remarkable 42% increase in Kappa scores. It's worth noting that the lowest Jaccard Score is observed for both the STFT cases. For the RAW case, the training accuracy starts at 92.3%, which slightly drops to 90.90% in the validation set and further decreases to 89.99% in the testing set. A similar trend is observed for the NORM dataset as well. In the case of spectral analysis, however, the validation set records the lowest values across the evaluation matrices. The model's output being a softmax cell, it is notable that both the testing accuracy and the testing Cosine Similarity matrices exhibit consistent and coherent trends. These findings lead us to the critical conclusion that the model does not overfit the data. While a few data points may exhibit deviations, a t-SNE visualization, as depicted in Figure 4.9, along with an analysis of the internal representation of the data within our network, align with our expectations for training artificial neural networks. In Figure 4.9, the analysis reveals that all three phases of the model (training, validation, and testing) create clearly separated clusters in the representation of the training data. The first row of the figure represents the output of the Bi-LSTM block, which consists of feature sizes with 750 units. Remarkably, we can clearly distinguish the three distinct clusters for the training, validation, and testing data.

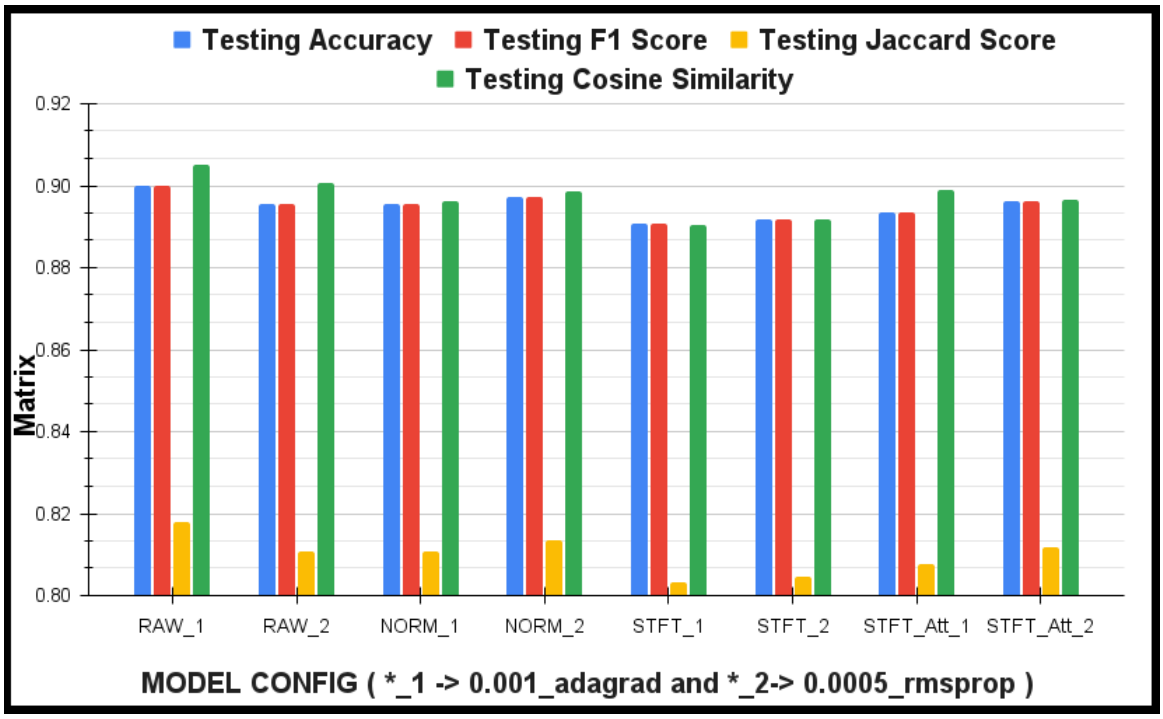


Figure 4.8: Bar graph of different evaluation matrices (X-axis) for different matrix of the parametric models tested (Y-axis).

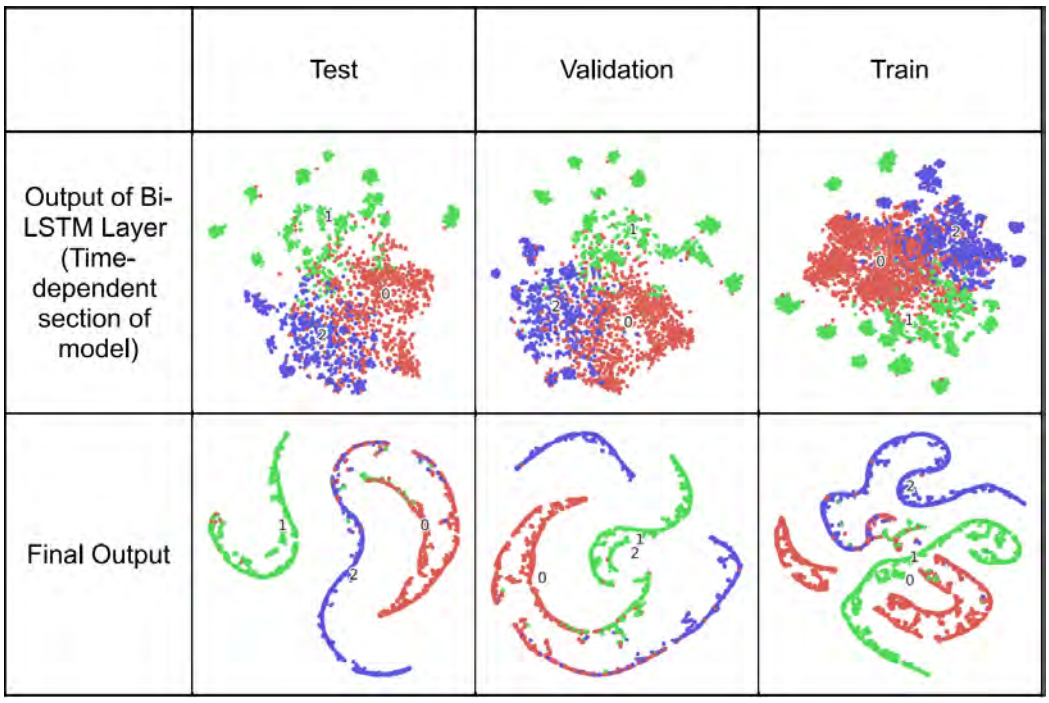


Figure 4.9: T-distributed stochastic neighbor embedding (t-SNE) embedding plots of the final output and the output of the BiLSTM layers with raw_0.001_Adagrad. Here, Red, Purple and Green colours stand for 'W', 'S' and 'MSE', respectively.

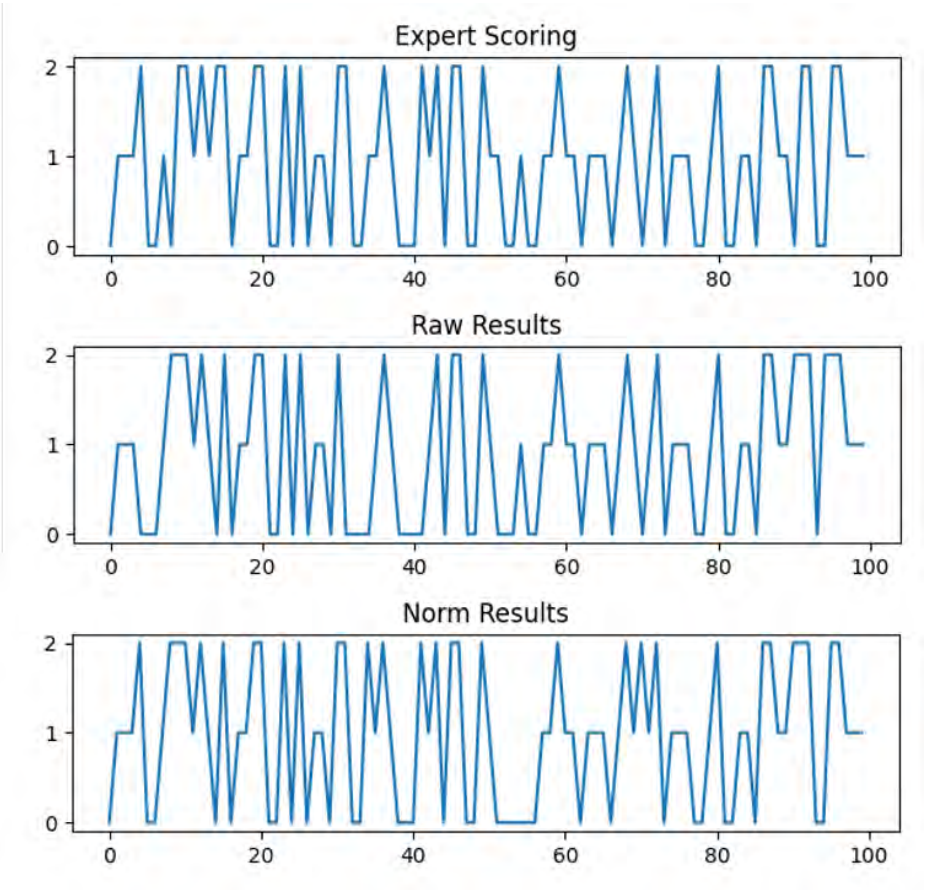


Figure 4.10: Temporal Analysis Output. Here, timestamps are shown on the x-axis and labels are depicted on the y-axis.

As mentioned in the previous section, our temporal analysis yields excellent results, which is also evident from the 50-minute long output Hypnogram for the RAW and the NORM cases. Figure 4.10 illustrates the baseline results alongside the predicted hypnograms. In most cases, an almost identical pattern is replicated for classes belonging to wakefulness and microsleeps (MSEs), with a smooth transition between them. Furthermore, our model's performance is exceptional in terms of Kappa values, especially when compared to the 16s-CNN and 16s-CNN-LSTM models referenced in [4]. The improvement in Kappa values for the Wakefulness and MSE classes ranges from 30% to 35%, signifying a significant advancement in these areas. Notably, we observe a remarkable 125% improvement in the detection of Deep Sleep. This substantial progress is particularly noteworthy given the limited number of training labels available for this

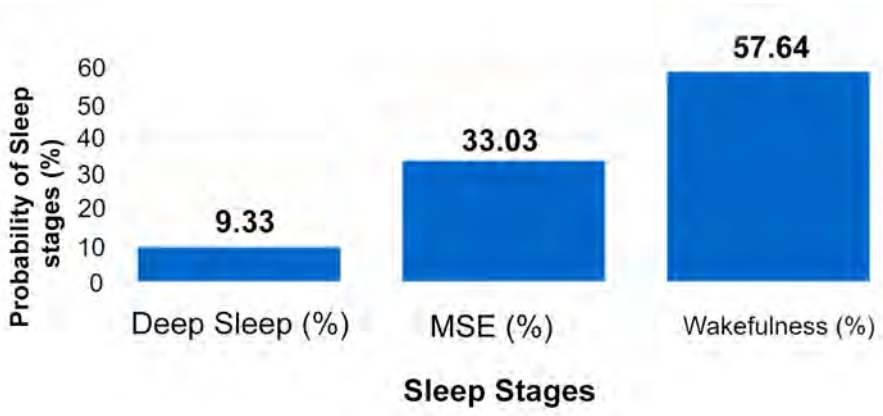


Figure 4.11: Microsleep Detection Results represented on User Interface

stage, highlighting the robustness and effectiveness of our model. The predictions generated by the model are seamlessly presented in real-time through a user interface designed with precision using the Streamlit framework, as depicted in Figure 4.11.

4.8 Summary

This chapter introduces an innovative hardware pipeline and a dedicated approach for real-time microsleep detection tailored for ADAS. The proposed methodology harnesses the NeuroSky MindWave Mobile EEG device to capture individual brainwave signals, seamlessly transmitting them to the Jetson Nano platform using Bluetooth technology. On the Jetson Nano, proposed well-established machine learning-driven microsleep detection model, enriched by Wavelet transform and Short Time Fourier Transform (STFT), adeptly classifies input data into "wakefulness," "Microsleep (MSE)," and "Deep Sleep" states. The system showcases a remarkable level of precision in real-time, with results accessible through an intuitive web interface. This research represents a significant advancement in the realm of microsleep detection for ADAS, where the rapid and accurate identification of microsleep events holds paramount importance. The proposed approach promises to enhance the safety and reliability of ADAS by proactively identifying

and addressing driver fatigue and drowsiness, ultimately contributing to safer road experiences.

Chapter 5

The Future of ADAS-Enabled Vehicles at Intersections with Traffic Congestion: Federated Learning and Vertical Partitioning Insights

5.1 Introduction

Intersections constitute critical junctures within urban traffic systems, where various streams of vehicles and pedestrians intersect, diverge, and contend for the right of way. The challenge of traffic congestion is particularly pronounced at these focal points, arising when the volume of traffic exceeds the intersection's capacity to manage it effectively [24–26]. Ineffectual traffic management at intersections results in vehicles coming to a standstill or proceeding at an exceptionally sluggish pace. This bottleneck effect can trigger a ripple effect, giving rise to traffic gridlock that extends well beyond the immediate vicinity of the intersection. The uninterrupted flow of traffic through intersections is of paramount importance, as

it ensures the safe and efficient traversal of vehicles and pedestrians. However, traffic congestion disrupts this essential flow, resulting in extended travel durations, heightened fuel consumption, and elevated stress levels for commuters. Mitigating traffic congestion at intersections stands as a pressing concern, with the potential to greatly enhance the quality and efficiency of urban transportation systems.

The presence of traffic congestion at intersections is attributed to a multifaceted interplay of factors. Predominantly, this issue arises when the number of vehicles seeking passage through an intersection exceeds the intersection's inherent capacity, resulting in traffic bottlenecks and consequent delays. Traditional fixed-time traffic signal systems, operating on predetermined intervals, often fall short in accommodating the dynamic and real-time traffic conditions [28]. These systems may inadequately respond to fluctuations in traffic loads, compounding congestion challenges. Human behavior constitutes another vital facet in the genesis of congestion. Driver actions, encompassing reckless driving, non-compliance with traffic regulations, and impulsive decision-making, can exacerbate traffic bottlenecks [168, 169]. Moreover, idling a car at intersections exerts an influence on the driver's physiological well-being. A study by Bou *et al.* [27] has highlighted the link between idling and changes in a driver's systolic and diastolic blood pressure. The relevance of these blood pressure parameters to drivers and traffic management is underscored by their potential impact on an individual's driving abilities [170]. Fluctuations in systolic and diastolic blood pressure can manifest through various symptoms, including headaches, dizziness, fatigue, and vision problems [27]. These symptoms can significantly compromise a driver's alertness, concentration, and reaction time, rendering them more susceptible to accidents and errors on the road. Therefore, understanding the physiological implications of idling at intersections becomes imperative not only for traffic management but also for ensuring the safety and well-being of drivers and road users. Further-

more, the reliance on outdated or historical traffic data for traffic planning, instead of real-time data, may result in suboptimal traffic management strategies. Neglecting the real-time traffic milieu can lead to subpar signal timings and traffic flow management. Contemporary traffic signal systems are grappling with inefficiencies when confronted with the burgeoning vehicular influx on modern roadways. The conventional traffic signals, which have dutifully served urban areas for many years, are now finding themselves ill-equipped to address the demands of today's dynamic and rapidly evolving urban environments. Fixed-timed signals, once the cornerstone of traffic management, are increasingly challenged to effectively navigate the ever-fluctuating traffic patterns at intersections. The consequence of this inability is a surge in delays and inefficiencies that impact the overall flow of urban transportation. Adding to the complexity, these traditional traffic signals adhere to pre-determined schedules, lacking the adaptability required to respond to real-time conditions. This deficiency further exacerbates the mounting issue of traffic congestion. As urban traffic continues to evolve, there is a growing imperative for innovative and adaptive traffic signal systems that can adeptly manage the complexities of contemporary traffic scenarios.

Intersections pose numerous challenges and complexities for both traditional vehicles and ADAS-enabled cars. Traditional vehicles necessitate manual operation by drivers, relying on human observations and reactions to navigate these intersections. This heavy reliance on human judgment can lead to delays and, in congested conditions, even accidents. Lacking the advanced sensors and cameras characteristic of ADAS, traditional vehicles exhibit limited situational awareness regarding their surroundings. This limitation can prove challenging when identifying potential obstacles or adapting to sudden shifts in traffic patterns. Another constraint is the incapacity of traditional cars to share real-time traffic data with central systems, hindering the optimization of signal timings and traffic management. In stark contrast, ADAS-equipped vehicles present several notable advan-

tages when it comes to intersection navigation. These vehicles are equipped with an array of sensors and cameras that provide real-time data about their immediate surroundings [29, 30]. This heightened situational awareness enables them to detect potential obstacles, pedestrians, and other vehicles at intersections, empowering them to take more proactive actions. Many ADAS-enabled vehicles offer semi-autonomous features, including adaptive cruise control, lane-keeping assistance, and automated emergency braking. These features provide substantial support to drivers in safely navigating complex intersection scenarios, thereby diminishing the risk of accidents. Moreover, ADAS-enabled vehicles have the capability to communicate with intelligent traffic signals, which are equipped with cameras and sensors [29, 30]. This communication facilitates the receipt of signal information in advance, allowing for smoother deceleration and acceleration, ultimately reducing stop-and-go traffic. In certain instances, ADAS systems within vehicles can share crucial traffic data with central traffic management systems. This data-sharing mechanism contributes to more effective traffic management and the fine-tuning of signal timings, leading to a reduction in congestion [31]. Within the realm of congested intersections, the distinction between ADAS-enabled vehicles and traditional cars becomes evident. ADAS-equipped vehicles leverage their advanced technology to enhance safety, improve traffic flow, and contribute to more efficient urban mobility. As the prevalence of ADAS systems continues to grow, their significant impact on intersection dynamics and traffic congestion is poised to reshape the landscape of urban transportation.

Amidst a global population of one billion cars, a mere 10% are equipped with ADAS technology [11]. This discrepancy leaves a significant portion of drivers and passengers without access to the comprehensive benefits offered by ADAS, especially at intersections. Consequently, a robust solution is needed, one that can be implemented at intersections or traffic signal nodes within a VANET environment, providing utility to both ADAS-enabled systems and traditional vehicles.

The key to addressing congestion at intersections lies in the deployment of intelligent traffic signals. These smart signals leverage cutting-edge technologies, including camera systems, to dynamically adapt to real-time traffic conditions. They diligently monitor each lane, track vehicle counts, and assess traffic density. Employing innovative techniques such as Federated Learning and Vertical Partitioning, these signals establish intercommunication and data sharing across the intersection network. This collaborative intelligence empowers the system to optimize signal timings, ultimately reducing wait times and enhancing the overall traffic flow. With the allocation of green and red times guided by this intelligence, the system ensures seamless passage for both vehicles and pedestrians while minimizing idling and emissions. The future of ADAS-enabled vehicles at congested intersections is intrinsically linked to the adoption of this intelligent traffic signal solution. It heralds the prospect of a more sustainable, efficient, and streamlined urban mobility landscape, wherein the benefits of ADAS are extended to a broader spectrum of vehicles and commuters.

This chapter introduces a novel approach to address the issue of idling at red lights and enhance the fuel efficiency of Intelligent Transport Systems (ITS) through Privacy-Preserving Asynchronous Federated Learning and Vertical partitioning. The key contributions of this chapter are as follows:

1. The idling problem was effectively addressed by classifying it into two distinct clusters: primary and secondary clusters, providing a structured approach to understanding and mitigating idling issues.

2. The proposed algorithm demonstrated practicality by considering multiple factors contributing to resistance, including cumulative headway start time, physical characteristics, and vehicle-specific coefficients. This approach enhances the algorithm's applicability to real-world scenarios and specific vehicle specifications.

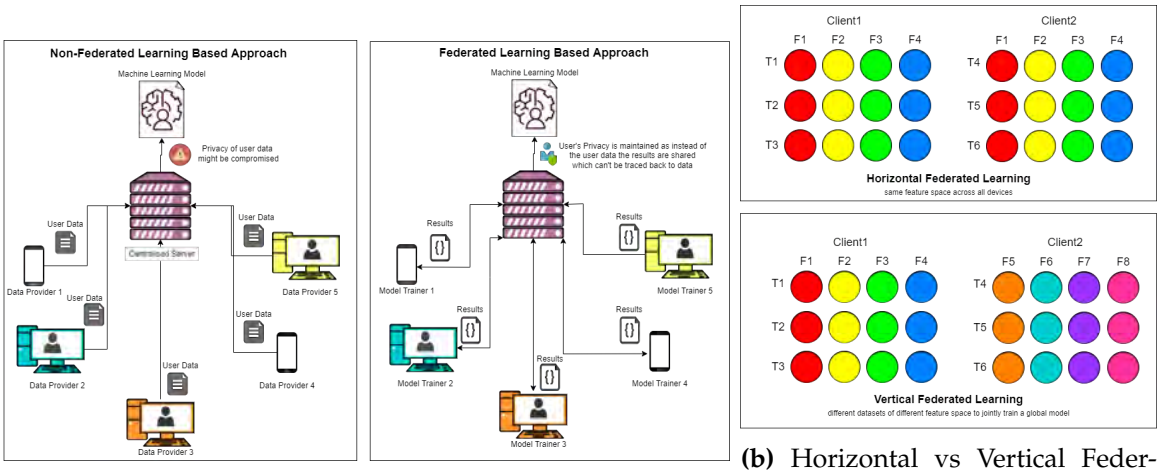
3. Notably, the proposed technique significantly improved the passing vehicle ratio, elevating it from 0.88 to 1.33. This enhancement implies a simultaneous passage of a greater number of vehicles at the intersection, ultimately leading to improved traffic flow and reduced wait times.

4. The chapter introduced a mathematical model that effectively illustrates the relationship between travel time and fuel consumption. Emphasizing the importance of reduced idling, the model underscores the potential for decreased fuel consumption and reduced carbon emissions, aligning with sustainability and environmental goals.

5.2 Proposed Model

Numerous contemporary approaches to traffic signal control and traffic forecasting rely on the utilization of user-specific information, including device GPS data and, in some cutting-edge instances, users' social media data, as a means to enhance their capabilities in managing traffic flow [171]. However, the deployment of such techniques inadvertently encroaches upon user privacy. To address this issue, various privacy-preserving regulations and legislations have been drafted, such as the European Union's General Data Protection Regulation [172] and the Cyber Security Law of the People's Republic of China, aimed at curbing unauthorized data collection to protect individual privacy [173].

In the context of machine learning (ML), traditional algorithms heavily rely on large-scale training data to attain accuracy levels akin to human performance, often involving the use of sensitive personal information. However, due to the aforementioned privacy regulations, the collection of personal data is subject to strict limits. These regulations, while safeguarding individual privacy, also present a challenge by impeding the progress of ML and AI research. When datasets encompassing diverse features are distributed across multiple data holders within



(a) Federated vs Non-Federated Learning

(b) Horizontal vs Vertical Federated Learning

Figure 5.1: Pictorial Comparison.

the same entity, such datasets are referred to as "vertically partitioned datasets." While integrating these data can potentially yield more powerful models and improved outcomes, direct data sharing is not viable due to the sensitive and private nature of the information. One promising approach to circumvent these challenges is Federated Learning (FL). FL represents a novel paradigm in which the primary objective is to enable users to perform computations on their data privately, obtain corresponding results, and share only these results, rather than the underlying user data, with other peers or servers until the desired level of accuracy is achieved. Moreover, FL ensures that the original dataset cannot be easily reconstructed from the shared results, rendering it a more secure alternative to the traditional approaches. The distinction between Federated Learning and non-federated existing solutions is visually depicted in Figure 5.1a.

Additionally, within the realm of Federated Learning, there exist two prominent approaches: Horizontal Federated Learning, which involves combining the same features from different clients, and Vertical Federated Learning, which focuses on specific entities and creates a unified model from features gathered from diverse clients. This distinction is visually elucidated in Figure 5.1b. From an optimization perspective, Wan et al. [174] have devised a privacy-preserving solution

tailored for vertically partitioned (VP) data. Zhang et al. [175] have introduced an FD-SVRG model designed for higher-dimensional linear classification. However, it's crucial to note that these Federated Learning methods, especially those based on vertically partitioned data sets, often rely on synchronous computing, where all processes operate using the same clock, or tasks are executed sequentially. This synchronous approach can be inefficient, as it may lead to idle devices wasting valuable resources and slowing down the overall process. In contrast, asynchronous computation, where processes have independent clocks or multiple tasks can be concurrently scheduled on a single thread, offers a more resource-efficient and expedited alternative.

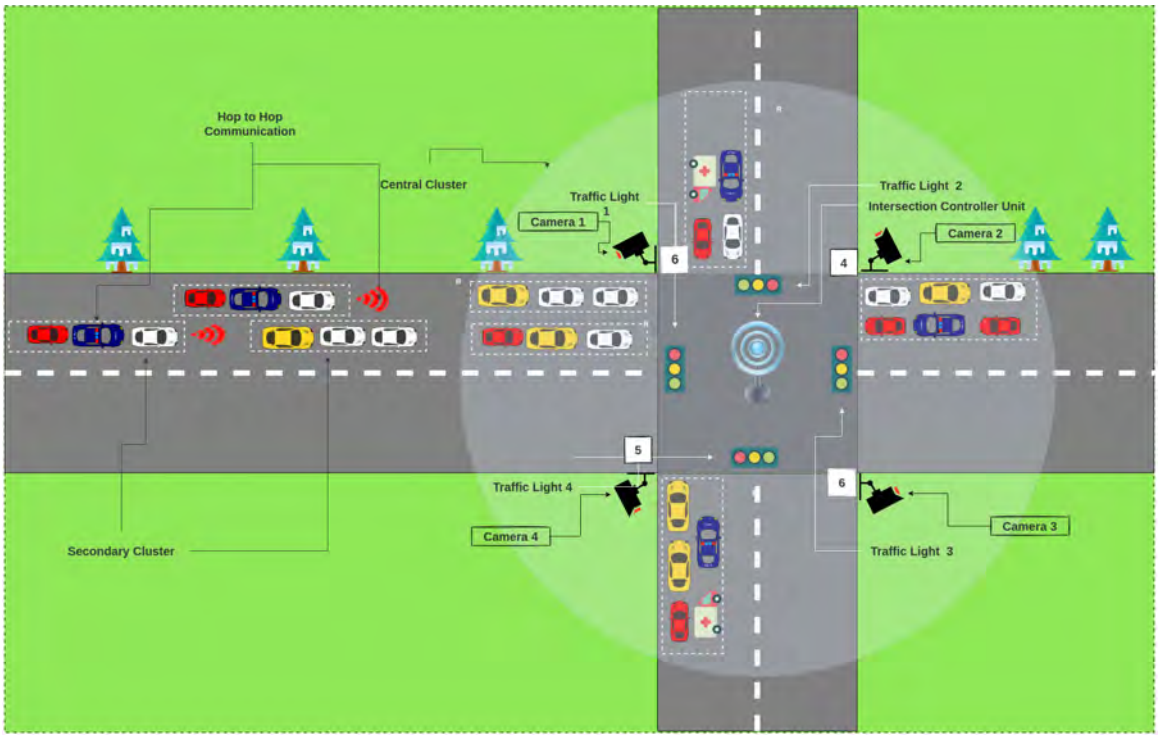


Figure 5.2: Scenario for Proposed Model

The elemental concept revolves around implementing adaptive acceleration and deceleration strategies contingent on traffic congestion levels in the vicinity of red traffic signals, while optimizing traffic signal schedules. This entails the development of various algorithms tailored to specific scenarios, dictated by factors such

as traffic density, time of day, location, and environmental variables like weather conditions and road infrastructure [176, 177]. Figure 5.2 provides a visual representation of the proposed model's scenario. The core objective of this model is to curtail emissions, conserve fuel resources, diminish the carbon footprint, and ensure the sustainable utilization of resources for future generations. For the purposes of this illustration, we assume a scenario where traffic density remains relatively consistent on all four sides of the traffic signal. Also, we've made certain assumptions which are:

1. The round-robin algorithm is employed to control the transition of traffic signals between red and green lights, ensuring an equitable distribution of signal changes across all sides with minimal risk of any side experiencing prolonged signal starvation.

2. Cameras pre-installed in each lane of the traffic light diligently monitor the surrounding area, providing comprehensive surveillance coverage.

3. A cluster-based Hop-to-Hop Communication system is implemented for vehicles located beyond central cluster. This allows vehicles to share real-time information regarding traffic congestion with the central cluster, enabling efficient traffic management.

5.2.1 Central Cluster and Secondary Cluster

The area proximate to the traffic light is designated as the central cluster. All four intersections converge at this central junction, necessitating the installation of cameras to monitor each lane. Consequently, for each intersection, a total of four cameras are strategically positioned. These cameras have a range equivalent to the radius of the central cluster, capturing real-time views of their respective lanes, as shown in figure 5.2. To detect vehicles, the YOLO (You Only Look Once) model [178] is employed due to its commendable accuracy and efficiency. The detected

vehicles are further categorized into various classes, enabling the assessment of traffic density and the monitoring of vehicle counts for each lane. Each vehicle is individually tracked using its centroid.

The duration of the green signal time is determined based on the observed traffic volume. Denser traffic areas are allocated a longer green signal time than areas with lighter traffic. This process is repeated for every lane within the central cluster, and the signal lights are updated accordingly. Concurrently, a secondary cluster has been introduced, positioned outside the central cluster. Its primary purpose is to ensure the safety and comfort of passengers and drivers by gradually decelerating vehicle speeds and eventually coming to a full stop. In situations where congestion arises in the central cluster, the secondary cluster is alerted. Subsequently, the secondary clusters establish hop-to-hop communication to facilitate speed reduction and minimize idling time, guided by a probabilistic approach. This integrated system aims to enhance traffic flow, safety, and user experience in urban environments.

5.2.2 Detection of Vehicles

To effectively detect and analyze vehicles for assessing traffic density, the proposed system employs the YOLO (You Only Look Once) model [178]. YOLO is a state-of-the-art algorithm for real-time object detection and tracking, making it ideal for vehicle detection from camera footage. A customized YOLO model is used to recognize various vehicle classes, categorizing them based on passenger capacity and physical attributes. These vehicle classes include Trucks (heavy logistics vehicles), Cars (personal passenger vehicles in the Light Motor Vehicle category), Auto-Rickshaws (service-based passenger vehicles or commercial vehicles), and Bikes (Motorcycle category).

For training the YOLO model, Google photos are manually annotated using La-

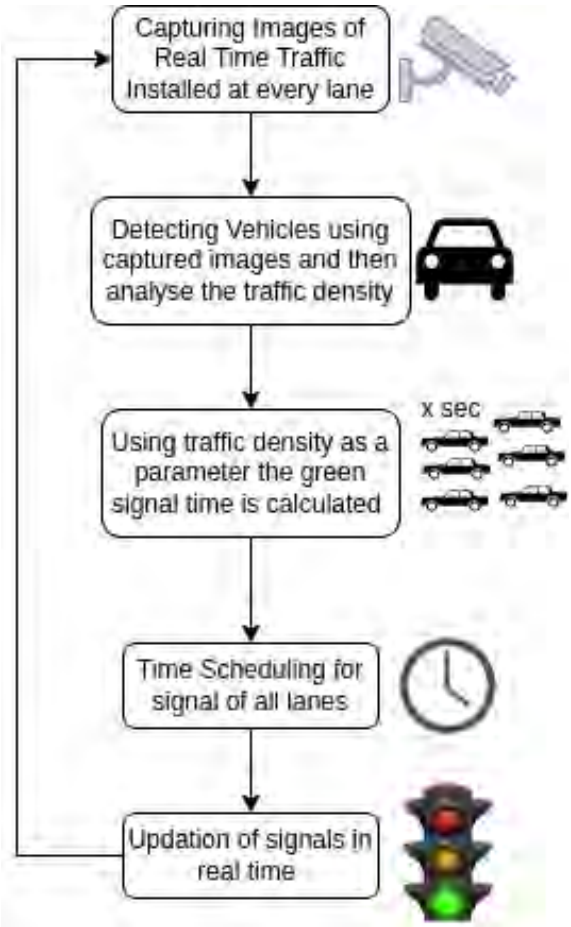


Figure 5.3: Flowchart of proposed model

belIMG [179]. The model is trained until the loss significantly decreases and stabilizes, ensuring its accuracy. Subsequently, the trained model is used to analyze test images, providing vehicle recognition outputs in a JSON format that includes labels, confidence scores, and coordinates. The YOLO algorithm is a robust choice for vehicle detection due to its high precision, low latency, and real-time object detection capabilities, which are essential for accurate traffic density assessment. Figure 5.3 provides a visual representation of the workflow employed to predict the density of vehicles per lane and subsequently determine the Green Signal Time (GST). This flowchart offers a clear depiction of the sequential steps and decision points involved in the proposed model, highlighting its overall process for efficient traffic management.

5.2.3 Signal Timing and Switching Algorithm

The proposed algorithm determines the Green Signal Time (GST) for a specific lane at a given moment. Additionally, it manages the cyclic switching of green lights across multiple lanes and concurrently updates the red light duration for other lanes. To ensure the process operates efficiently and seamlessly, multi-threading is employed, allowing for asynchronous execution and significantly faster processing. The algorithm takes input from the vehicle detection module, utilizing the data on the number of vehicles in each vehicle class. Several factors are considered in the algorithm's design, including traffic density, the vehicle class with the highest number of vehicles, the number of lanes, the average vehicle speed, the time required for vehicle startup, the exponential waiting time for other vehicles in the queue, algorithm processing time, and the timing for capturing snapshots. These considerations are essential for optimizing traffic signal timings, reducing congestion, and improving traffic flow.

The Green Signal Time (GST) is directly influenced by traffic density. Increased traffic density corresponds to a longer required GST. The type of vehicles plays a crucial role in this parameter, as heavy vehicles like trucks or buses significantly contribute to traffic density, requiring longer GST than lighter vehicles such as bikes or cars. The time at which a snapshot needs to be taken is determined by the time required for running the algorithm. When the signal switches, the Green Signal Time (GST) of the next signal must be kept ready by that time so that there is minimal lag in switching. Assuming vehicle detection, traffic density estimation, and Green Signal Time (GST) calculation takes 5 seconds, then we need to take a snapshot as soon as the green light turns yellow for any lane; then we'll have 5 seconds to predict the duration of GST for next lane. This ensures that there is sufficient time to determine the GST for the next lane, minimizing switching delays. The algorithm switches traffic signals in a cyclic manner rather than

favoring the densest lane. This approach prevents situations of lane starvation, where lanes with lower traffic density experience excessively long waiting times. Instead, the GST dynamically adjusts to the traffic rate on each lane. In this way, the algorithm aims to reduce the average idling time of vehicles. The duration needed for different classes of vehicles may depend on various factors such as the locality, road type, and other parameters. These values can be modified to further reduce average idling time and enhance traffic flow.

In the case of an n-lane intersection, where n equals 4, each lane is designated with a specific label for clarity. The lane heading rightwards is denoted as l_0, the lane going downwards is represented as l_1, and so forth in a cyclic sequence. Since we have taken into account vehicles of four distinct classes, it's essential to have data regarding the number of vehicles, denoted as $count_{class}$, belonging to each class, along with the adjusted average time, labeled as $adjAvgTime_{class}$, required for a vehicle of that class to cross the intersection.

$$adjAvgTime_{class} = avgTime_{class} + avgTimeStartup_{class} \tag{5.1}$$

$$\begin{aligned} AllClassAverageTime = (count_{truck} * adjAvgTime_{truck}) + \\ (count_{car} * adjAvgTime_{car}) + \\ (count_{Auto} * adjAvgTime_{Auto}) + \\ (count_{bike} * adjAvgTime_{bike}) \end{aligned} \tag{5.2}$$

$$GreenSignalTime(GST) = \frac{AllClassAverageTime}{num_{lanes} + 1} \tag{5.3}$$

In the initial iteration, the lane l_0 is assigned a default value. Subsequently, the formula for calculating the Green Signal Time (GST) is applied.

5.3 Formulation of Mathematical Equations

Consider N lanes of the intersection, which are identified as $\zeta = 1, 2, \ldots, N$, in order. Let i be the lane-shifting counter. The upcoming lane shifts in a horizon of i steps which can be represented by the vector $\gamma(i)$, whereas γ = [γ_1, γ_2, . . . , γi] with its elements belonging to γ_j ϵ 1, 2, . . . , N and $j = 1, 2, ..., i$. For ease of use, the signal with green lights at the j^{th} step in the horizon is also referred to as γ_j. The total number of classes of vehicles ranges from $1, 2, \ldots, V$. The class of vehicles in consideration is given by the vector $\vartheta(v) = [\ \vartheta_1, \vartheta_2, \ldots, \vartheta_v]$ where v ranges from $1, 2, \ldots V$, and ϑ_v is the name of the category of any particular vehicle. The number of vehicles that crosses the intersection in the respective intervals is specified by $\beta \triangleq$[β_1, β_2, . . . , βi] ^{T}, where β_i is a vector representing the count of class-wise distribution of vehicles, $\beta_i = [\ \beta_{i1}, \beta_{i2}, ..., \beta_{iv}]$ and is a major factor in the determination of lengths of green signals in the horizon also referred to as Green Signal Time (GST). The average time required to pass the intersection is different for varying classes of vehicles, so an averaged value is considered based on the class of vehicle represented by the vector $\kappa(v) = [\ \kappa_1, \kappa_2, ..., \kappa_v]$, where κ_v is the averaged time required by the vehicle belonging to class $\vartheta(v)$ to cross the intersection. The startup of the vehicle also takes a certain time which is represented in a classwise manner by $\delta(v)$, where $\delta(v) = [\ \delta_1, \delta_2, ..., \delta_v]$, where δ_v is the averaged time required to start the vehicle belonging to class $\vartheta(v)$. So, the GST for the i^{th} interval $(\eta(i))$ can be calculated by multiplying the distance vector $\beta(i)$ with the summation of averaged time vector $\kappa(i)$ and the startup time vector $\delta(i)$, which is mathematically represented as,

$$\eta(i) = \frac{[\beta(i)^T_{(i\times v)}.(\kappa(v) + \delta(v))]_{(i\times 1)}}{(N+1)} \tag{5.4}$$

To cover multiple cycles, the horizon must have $i \geq N$. For instance, the lane

phases can be expressed as (i) = [1, 2, 3, 4, 1] for step i in an intersection with four lanes (N = 4), as in Fig. 5.2, and a five-step horizon (i = 5). For the given lane sequence $\gamma(i)$, the corresponding $\beta(i)$ is the class-wise input vector of the traffic signal control problem at some step i, and the actual switching times of the traffic lights (green, yellow or red of each section) are determined based on both the vectors $\gamma(i)$ and $\beta(i)$. The duration of GST is termed as $\eta(i)$, i.e., GST in i^{th} iteration on lane $\gamma(i)$. The starting and ending time of the green length of signals on lane $\gamma(i)$ are represented by the vectors $\xi(i) = [\ \xi_1(i), \xi_2(i), \ldots, \xi_i(i)]$ and $\varsigma(i) = [\ \varsigma_1(i), \varsigma_2(i), \ldots, \varsigma_i(i)]$. The green phase of lane $\gamma_1(i)$ begins with the horizon taken into account, i.e.,

$$\xi_1(i) = 0 \tag{5.5}$$

and ends at

$$\varsigma_1(i)\& = \frac{[\beta_1(i)^T.(\kappa(v) + \delta(v))]_{(i\times 1)}}{(N+1)} + \xi(i)\& = \eta_1(i) + \xi(i) \tag{5.6}$$

The timestamp for subsequent $\xi_i(i)$ and $\varsigma_i(i)$ for various lane phases i.e. $\gamma_2(i)$, $\gamma_2(i), \ldots, \gamma i(i)$ can be determined by the adding the ending time of previous lane phase and the time required to cross the intersection, following is the expression:

$$\xi_{j+1}(i) = \varsigma_j(i) + \kappa_v, \tag{5.7}$$
$$whereas\ j = 1, 2, ..., i-1$$

similarly, the ending time could be represented as a sum of starting time as well as the green signal time for that duration

$$\varsigma_{j+1}(i) = \xi_{j+1}(i) + \eta_{j+1}$$

The components of $\gamma(i)$, or a fixed cyclic series of lane phases, are easily repre-

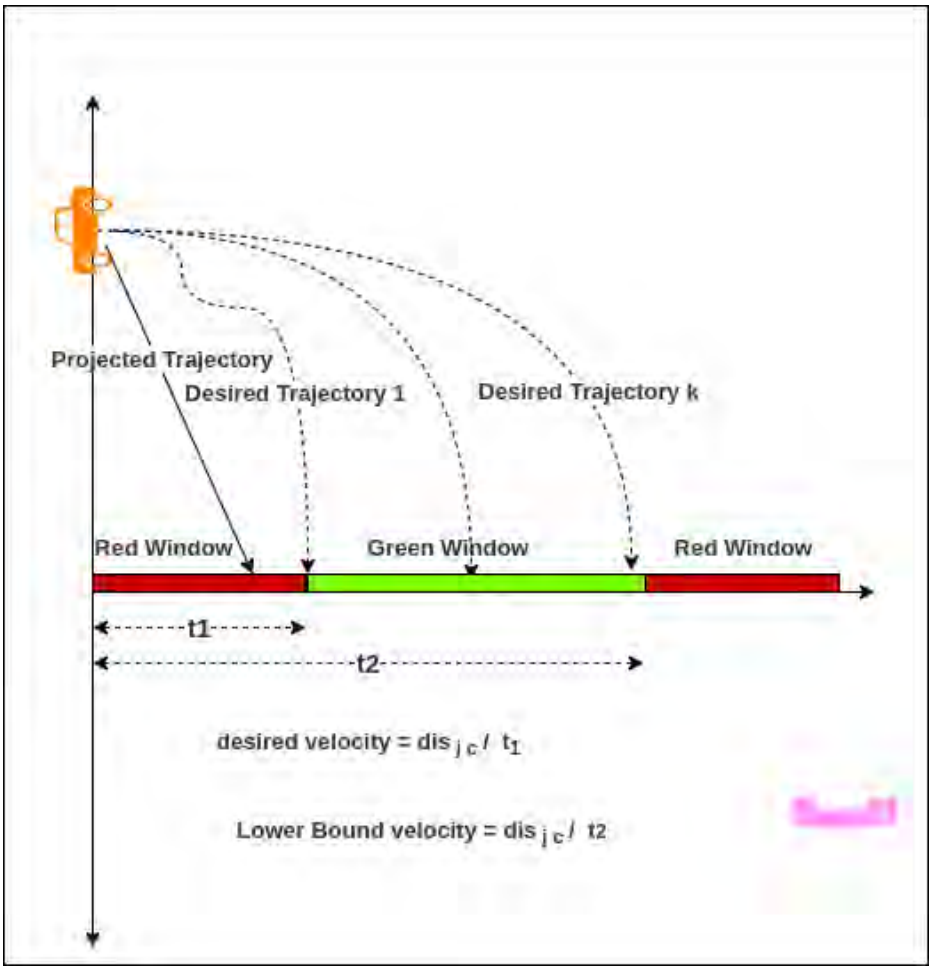


Figure 5.4: Determination of the desired velocity along with the representation of possible trajectories

sented as

$$\gamma_{j+1}(i) = \begin{cases} 1 & \text{if } \gamma_j(i) = N, \\ \gamma_j(i) + 1 & \text{otherwise} \end{cases} \tag{5.8}$$

5.3.1 Secondary Cluster Algorithm

To reduce the duration of idling time at the red light, the targeted velocity can be set based on the signal phase and time information rather than being set as the upper-speed limit on the road. According to [180], the following formula is used to determine each connected vehicle's desired speed at each iteration interval t

[181]:

$$\psi^i_{target}(t) = \begin{cases} \frac{dis_{ic}(t)}{\varphi_t - w_{green} - t} & \text{if } red, \\ \psi_{max} & \text{if green and } \frac{dis_{ic}(t)}{\varphi_t - w_{red} - t} \leq \psi_{max}, \\ \frac{dis_{ic}(t)}{\varphi_t + w_{red} - t} & \text{if green and Otherwise .} \end{cases} \tag{5.9}$$

$$w^i_{cycle} = w^i_{green} + w^i_{red} \tag{5.10}$$

$$\varphi_t = \sum_{i=1}^{t} w^i_{cycle} \tag{5.11}$$

The above equations describe a system that calculates the targeted speed of a vehicle approaching a traffic signal. The distance between the vehicle and the traffic signal at a given time is denoted by $dis_{ic}(t)$. The green light window is represented by w_{green}, while the red light window is represented by w_{red}. The total cycle duration is represented by w_{cycle}. The targeted speed is calculated by dividing the distance between the traffic light and the duration remaining before the commencement of the green window for a particular cycle. If the car can complete passing through the intersection during the green window, then the speed threshold is taken as the target velocity when the traffic light is green. In that instance, the next green window is targeted with the chosen target velocity. In addition to the target velocity, which is aimed at the beginning of the green window, the lower bound of the velocity range is also calculated. The lower bound is calculated by targeting at the end of the green window rather than targeting at the target velocity. Figure 5.4 shows how to determine desired velocity along with possible trajectories.

5.3.2 Position Parameters and Fuel Optimization

The pros of Model Predictive Control (MPC) include its ability to handle bounded constraints and its ability to optimize the current time step while still considering future time steps [182]. The longitudinal dynamics of the vehicle j are taken into

consideration concerning the reference [183]. Table 5.1 describes various notations used to represent equations and algorithms. The variables s_j represent the position and v_j represent the velocity of the vehicle, given by:

$$\alpha_j = [\ s_j v_j]\ ^T \tag{5.12}$$

$$\hat{\alpha_j} = f_j(\alpha_j, u_j) \tag{5.13}$$

$$f_j(\alpha_j, u_j) = \left[v_j - \frac{1}{2M_h^j}.C_D.\rho_a.A_v^j.v_j^2 - \mu.g - g.\theta + u_j \right] \tag{5.14}$$

Here, u_j is the input parameter, i.e., the vehicle's braking force per unit mass j. Additionally, some other terms such as M_h^j represent the mass of the vehicle, C_D indicates drag coefficient. Along with that, ρ_a represents the air density, A_v^j is frontal area. Whereas μ is rolling friction coefficient, g is gravitational constant, and θ is road gradient in percentage. This section introduces the fuel consumption evaluation of a traditional car. The polynomial metamodel introduced in research [182] is used to assess the fuel consumption rate for a traditional car in this study.

$$\hat{\omega^j_{fuel}}(t) = \hat{\omega^j_{const}}(t) + \hat{\omega^j_{accl}}(t) \tag{5.15}$$

$$\hat{\omega^j_{const}}(t) = b_0 + b_1.v_1(t)^1 + b_2.v_2(t)^2 + b_3.v_3(t)^3 \tag{5.16}$$

$$\hat{\omega^j_{accl}}(t) = \hat{a_j}.\left(r_0 + r_1.v_1(t)^1 + r_2.v_2(t)^2\right) \tag{5.17}$$

$$\hat{a_j} = \frac{-1}{2M_h^j}.C_D.\rho_a.A_v^j.v_j^2 - \mu.g - g.\theta + u_j \tag{5.18}$$

In the aforementioned equations, the terms $\hat{\omega^j_{const}}(t)$ and $\hat{\omega^j_{accl}}(t)$ denote the fuel consumption rates of a vehicle moving at a constant speed and an accelerating vehicle, respectively. b_0, b_1, b_3, r_0, r_1, and r_2 are vehicle-specific constants [184]. If $v_j(t) = 0$ or $u_j < 0$, it denotes the halting or idle motion of the automobile. $\hat{\omega^j_{fuel}}(t)$ can be set to $\hat{\Omega^j_{idle}}$ to set the fuel consumption as constant for the sake

S.No.	Notation	Meaning
1	N	Total Number of lanes
2	i	Lane Shifting Counter
3	$\gamma(i)$	Lane which is currently green
4	V	Total number of classes
5	$\vartheta(v)$	vector of classes of vehicle
6	β	vector of number of vehicles crossing the intersection
7	β_i	class-wise count of vehicles crossing the intersection in ith iteration
8	$\kappa(i)$	vector representing average time to cross the intersection in a class-wise manner
9	$\delta(i)$	vector representing average time to start the vehicle in a class-wise manner
10	$\eta(i)$	length of green signal time in i th iteration
11	$\xi(i)$	starting time of green signal on lane gamma i in ith
12	$\varsigma(i)$	ending time of green signal on lane gamma i in ith
13	$\psi^i_{target}(t)$	vehicle's desired speed at each iteration interval t
14	α_j	longitudinal dynamics of the vehicle j
15	ω^j_{fuel}(t)	fuel consumption by the jth*vehicle* in t time

Table 5.1: Notations Used

of simplification. But in real-life scenarios, the polynomial function of a vehicle's speed and acceleration determines how much fuel it will need.

With the aforementioned way of evaluating fuel consumption rates, the fuel efficiency of any vehicle j at time t can be expressed as

$$\hat{\omega^j_{fuel}}(t)/v_i(t),$$

i.e., the units of fuel spent per unit of distance traveled. With a T-second time

horizon, the cost function of each vehicle j at each moment of time k is represented as follows [185] :

$$J_j(k) = \arg\min_{u_j} \sum_{j=k}^{k+T-1} \Bigg[w_1 \left(v_j(t) - v^j_{\text{target}}(k) \right)^2 + w_2^j(t) R_{ji}(t)^2 + w_3 u_j(t)^2 + w_4 \frac{\dot{\omega}^j_{fuel}(t)}{v_i(t)} \Bigg] \tag{5.19}$$

$$v^j_{lb}(t) \leq v_j(t) \leq v^j_{ub}(t), \quad \forall t \tag{5.20}$$

The target velocity derived from the centralized junction controller is penalized by the first component in the cost function (5.19). The second component is utilized to prevent crashes and maintain a predetermined desirable headway distance (S_0) and duration (t_{hd}) among the subject vehicle j and its preceding vehicle i, respectively (5.22). The last term reduces the fuel consumption rate per unit distance, which is the parameter via which our suggested technique will realize the improvement of fuel economy, while the third term minimizes the control effort. w_1, w_3 and w_4 are the corresponding constant weighting factors. $w_2(t)^j$ is chosen concerning the relative distance and is increasing as the distance grows closer and vice versa.

The constrained velocity and acceleration are shown in equation (5.20) and (5.21), $v^j_{lb}(t)$ is the lower bound of the velocity range that we previously mentioned, and $v^j_{ub}(t)$ represents our desired velocity, respectively. Figure 5.4 exhibits the determination of the desired velocity along with the representation of possible trajectories.

$$u^j_{lb}(t) \leq u_j(t) \leq u^j_{ub}(t), \quad \forall t \tag{5.21}$$

$$R_{ji}(t) = S_0 + t_{hd} \left(v_j(t) - v_i(t) \right) + \left(s_j(t) - s_i(t) \right) \tag{5.22}$$

5.4 Results

The model leverages the pygame framework to create a dynamic simulation, enabling an in-depth analysis of its statistical impact on real-world traffic scenarios. This simulation serves as a visual representation that not only aids in comprehending the model's structure but also sets it apart from the conventional traffic control systems employed in traffic lights.

The simulation features a 4-way junction, complete with four signal displays for vehicles. Positioned atop each signal is a timer that provides real-time information about the current signal status. This includes the time remaining before a green light transitions to yellow, a red light changes to green, or a yellow light shifts to red. Additionally, the simulation includes a tally of the number of vehicles that have traversed the intersection.

To construct this simulation, the model begins with a background image of a four-way intersection. This image is then manipulated to be displayed from various viewpoints, effectively emulating different directions. Based on the specific state of each signal—be it red, green, or yellow—a corresponding visual representation is generated using a designated code. The simulation is enriched with top-view images of various vehicles, such as rickshaws, buses, trucks, and bikes, alongside depictions of red, yellow, and green traffic lights.

The program incorporates an intelligent mechanism that takes into account the count of vehicles at each signal. This data is harnessed to calculate when a signal should transition, be it from green to yellow, red to green, or yellow to red, thereby ensuring a dynamic and responsive traffic control system.

For testing the efficiency of the model and sake of simplification, some values are substituted in the variables as N = 4 and V = 5, $\kappa(v) = [\ 1, 0.5, 1.25, 1.5, 1.5]$, $\vartheta(v) = [$ car, bike, rickshaw, truck, bus$]$

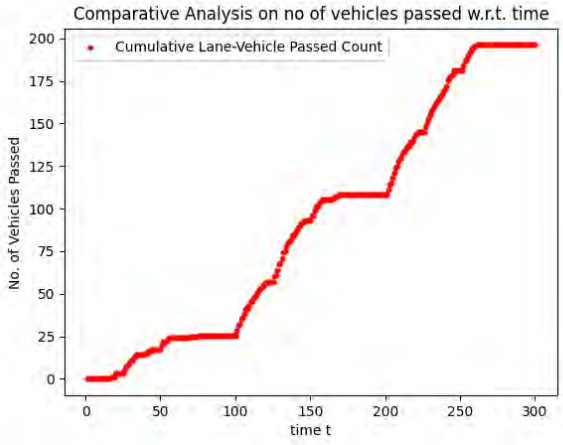
To replicate real-time traffic conditions, the simulation meticulously generates vehicles based on factors such as direction, lane, vehicle class, and turning behavior. The simulation allows for fine-grained control over the distribution of vehicles among the four directions, ensuring a realistic representation of traffic dynamics. New vehicles are dynamically created and introduced into the simulation at intervals of t = 0.75 seconds, with their attributes determined by random parameters assigned to their respective vehicle classes.

The simulation is intelligently programmed to model how vehicles respond to traffic lights, adhering to standard traffic rules. Vehicles halt in response to yellow and red signals and move forward when the signal turns green. Furthermore, a safe distance is maintained between vehicles; for instance, if a car is trailing a larger vehicle like a bus, it will reduce its speed to avoid collisions. To monitor and display critical simulation information, the concept of multi-threading is employed. This allows for the tracking of vehicles that have passed through the intersection and provides a real-time update on the elapsed simulation time. The simulation continues to run until the elapsed time matches the desired simulation duration. Despite the presence of just two sub-lanes in the image, an exclusive bicycle lane is introduced to the left of this setup. Vehicle turning behaviors are determined by random factors, enhancing the simulation's realism. For traditional traffic models, Figure 5.5a illustrates the ratio of passing vehicles concerning time, while Figure 5.5b displays the cumulative count of vehicles that have passed through each lane over time. Figure 5.5c offers a comparative analysis of the traditional model's performance. Similarly, for the proposed model, Figure 5.6a presents the ratio of passing vehicles in relation to time, and Figure 5.6b shows the cumulative count of vehicles that have traversed each lane over time. Figure 5.6c provides a comparative assessment.

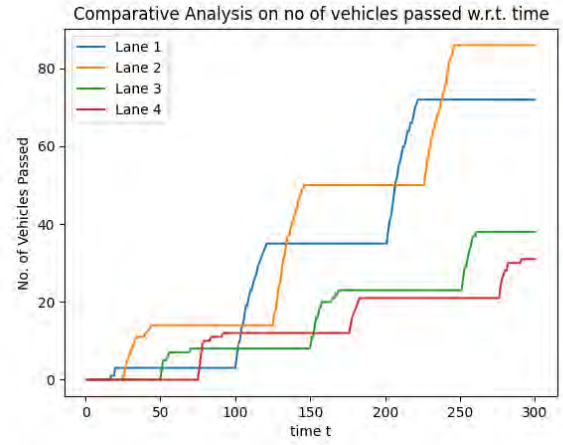
Table 5.2 provides insights into the ratio of passing vehicles to time across var-

```
   RED TS 1 -> r: 3  y: 5  g: 20
   RED TS 2 -> r: 103  y: 5  g: 20
   RED TS 3 -> r: 128  y: 5  g: 20
YELLOW TS 4 -> r: 0  y: 3  g: 0

   RED TS 1 -> r: 2  y: 5  g: 20
   RED TS 2 -> r: 102  y: 5  g: 20
   RED TS 3 -> r: 127  y: 5  g: 20
YELLOW TS 4 -> r: 0  y: 2  g: 0

Lane-wise Vehicle Counts
Lane 1 : 88
Lane 2 : 87
Lane 3 : 37
Lane 4 : 32
Total vehicles passed:  244
Total time passed:  300
No. of vehicles passed per unit time:  0.8133333333333334
```

(a) Ratio of passing vehicles with respect to time for Traditional models



(b) Cumulative lane-vehicles passed count with respect to time for Traditional models

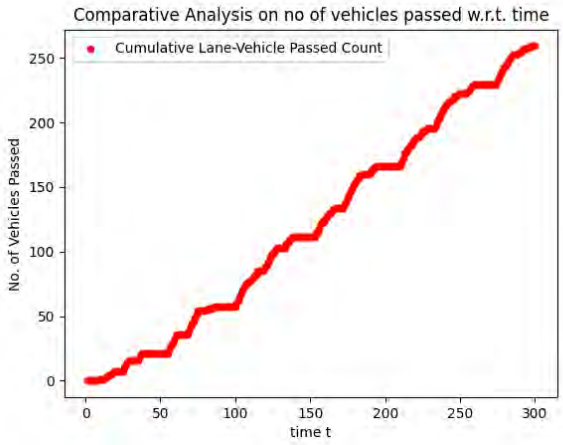


(c) Comparative Analysis on number of vehicles passed in each lane with respect to time for Traditional models

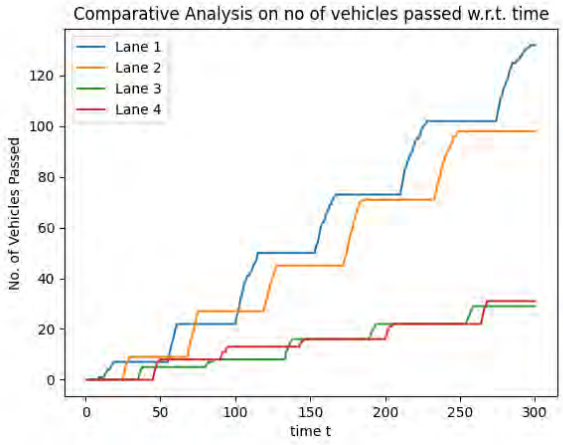
Figure 5.5: Results of traditional model

```
   RED TS 1 -> r: 142  y: 5  g: 20
 GREEN TS 2 -> r: 0  y: 5  g: 14
   RED TS 3 -> r: 19  y: 5  g: 20
   RED TS 4 -> r: 108  y: 5  g: 20

   RED TS 1 -> r: 141  y: 5  g: 20
 GREEN TS 2 -> r: 0  y: 5  g: 13
   RED TS 3 -> r: 18  y: 5  g: 20
   RED TS 4 -> r: 107  y: 5  g: 20

   RED TS 1 -> r: 140  y: 5  g: 20
 GREEN TS 2 -> r: 0  y: 5  g: 12
   RED TS 3 -> r: 17  y: 5  g: 20
   RED TS 4 -> r: 106  y: 5  g: 20

Lane-wise Vehicle Counts
Lane 1 : 171
Lane 2 : 151
Lane 3 : 36
Lane 4 : 39
Total vehicles passed:  397
Total time passed:  300
No. of vehicles passed per unit time:  1.3233333333333333
```

(a) Ratio of passing vehicles with respect to time for proposed model



(b) Cumulative lane-vehicles passed count with respect to time for proposed model



(c) Comparative Analysis on no of vehicles passed in each lane with respect to time for proposed model

Figure 5.6: Results of proposed model

S.No.	Traditional Ratio	Improvised Ratio
1	0.9933	1.3233
2	0.81	1.3422
3	0.8133	1.3122
4	0.9124	1.4011
5	0.8824	1.3102
Avg	**0.8828**	**1.3378**

Table 5.2: Ratio of passing vehicles to time in various iterations

ious iterations. This analysis leads to the conclusion that our proposed model outperforms traditional models, exhibiting a significantly improved ratio of 1.33 unit vehicles passing per unit time, in contrast to the traditional ratio of 0.88. Further enhancements can be explored by fine-tuning the model and adapting it to specific traffic conditions in the target area.

5.5 Summary

This chapter presents a novel framework for traffic congestion management at intersections by leveraging the power of Federated Learning and Vertical partitioning. The proposed technique addresses the issue of idling at red lights, leading to improved fuel consumption in ITS which includes ADAS enabled vehicles as well as other vehicles. Moreover, it demonstrates remarkable efficiency in enhancing the passing-vehicle ratio at intersections, resulting in more efficient traffic flow. The mathematical approach used in this study sheds light on the relationship between fuel consumption and travel time, underscoring the importance of reduced idling for fuel conservation and lower carbon emissions. The evaluation of our proposed model using various parameters confirms its effectiveness, with the passing-vehicle ratio reaching an impressive 1.33, surpassing the traditional model's ratio of 0.88. These findings highlight the significant advantages of our model over conventional vehicle passing per unit time ratio approaches, making it a promising solution for intelligent traffic management.

Chapter 6

Enhancing On-Road Perception for ADAS-Enabled Vehicles in Adverse Weather Conditions

6.1 Introduction

In the contemporary era, the advent of self-driving cars and ADAS has substantially enhanced human comfort and revolutionized transportation, delivering both speed, comfort and safety. However, these ADAS-enabled vehicles encounter significant challenges, particularly in adverse weather conditions. Among these conditions, hazy weather stands out as a critical environmental factor that severely hampers the object recognition capabilities of autonomous driving systems and ADAS-equipped vehicles. This impairment directly impacts the decision-making functionality of these vehicles, ultimately contributing to a heightened risk of accidents and injuries on the road.

In ADAS equipped vehicles, the reliance on cameras for various safety and assistance features is paramount. However, the efficacy of these features can be

severely compromised in hazy environmental conditions, leading to reduced vehicular visibility. Haze obstructs obstacle detection, road and lane recognition capabilities, significantly impacting the functionality and effectiveness of safety features like adaptive cruise control and lane-keeping assistance. The challenges posed by hazy atmospheric conditions further manifest in the form of misinterpretations by detection systems. These misinterpretations encompass both false positives, where non-existent objects are detected, and false negatives, where genuine obstacles go undetected. Such erroneous detections can result in unnecessary warnings or missed hazard alerts, potentially compromising road safety. In light of these challenges, there is a clear and pressing need for a state-of-the-art computer vision model capable of providing haze-free images. These clear images serve as a foundation on which various object detection and recognition models can operate effectively, enabling ADAS-enabled vehicles to navigate seamlessly in hazy environments. This chapter introduces the AGD-NET (Attention-Guided Dense Inception U-Net) model, which has been developed to address the critical issue of dehazing hazy images. Furthermore, it showcases the model's dehazing capabilities, specifically in the context of on-road hazy conditions, aiming to enhance the safety and functionality of ADAS systems in challenging environments. The contribution of this chapter is as follows,

1. Introduced AGD-NET, an attention-based encoder-decoder dehazing network, for effectively generating high-quality haze-free images.
2. AGD-NET utilizes the Attention-Guided Dense Inception U-Net architecture, enhancing its ability to capture and restore important features in hazy images.
3. AGD-NET achieves comparable performance compared to existing state-of-the-art methods in terms of PSNR and SSIM and establishes a new benchmark on the Dense-Haze dataset.

4. The evaluation on Dense-Haze, I-Haze, O-Haze, and NH-Haze datasets demonstrates the effectiveness of AGD-NET in removing haze and enhancing the quality of images.

5. In the realm of on-road dehazing capabilities, the AGD-NET model establishes a benchmark when compared to other models. It demonstrates its on-road dehazing capabilities for ADAS-enabled vehicles as well as self-driving cars.

By introducing AGD-NET and utilizing the Attention-Guided Dense Inception U-Net, the research significantly advances the field of image dehazing by improving the quality of haze-free images.

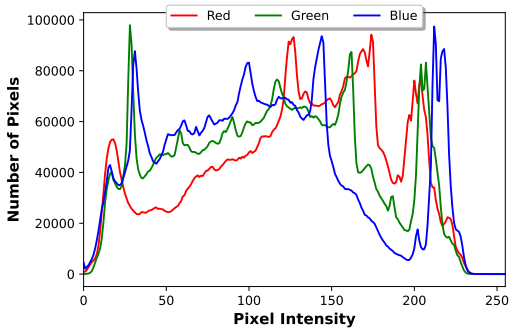
6.2 Background

6.2.1 Impact of Haze on Image Features

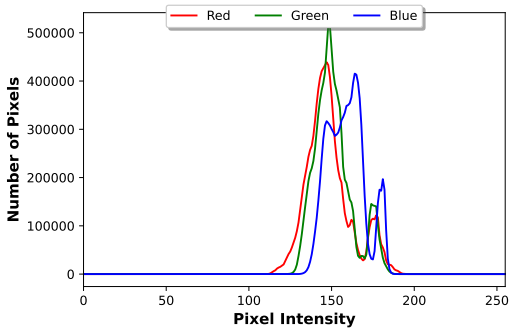
The image quality captured by a camera can be degraded in a hazy environment due to the absorption of moving particles present in the air. This degradation is characterized by a reduction in image contrast and color distortion, as discussed in [186]. Furthermore, the boundaries of objects and fine textures within the image become blurred, contributing to a loss of image clarity. To visually illustrate the impact of haze on image histograms, Figure 6.1 provides a compelling comparison. Within this figure, Figure 6.1a showcases an image captured in a normal, haze-free environment, while Figure 6.1c presents the corresponding histogram of this image. In contrast, Figure 6.1b displays an image captured under hazy conditions, and Figure 6.1d depicts the histogram associated with this hazy image. This clear differentiation in histograms underscores the significant disparities between images taken in normal and hazy environments, accentuating the adverse effects of haze on image quality. In the domain of image processing and the represen-

(a) Normal Environment [6]

(b) Hazy Environment [6]



(c) Histogram for (a)



(d) Histogram for (b)

Figure 6.1: Histogram comparison for normal and hazy image

tation of atmospheric phenomena, various methodologies have been devised to quantify the impact of fog or haze when capturing images. Two prominent techniques are the dark channel prior [112] and the grey system theory [187] methods.

The dark channel prior method focuses on assessing the density of haze within images. It is built upon the insightful observation that in clear outdoor images, many localized regions exhibit pixels that are exceptionally dark in one or more color channels. This method capitalizes on this observation to estimate the extent of haze, contributing to the enhancement of image quality and clarity in hazy conditions.

On the other hand, the grey system theory method leverages grey numbers and functions to address the uncertainty and complexity associated with fog and haze in weather conditions. These grey system tools are adept at capturing the interval-valued unknowns pertaining to how the atmosphere scatters and absorbs light. By

employing these tools, it becomes feasible to calculate important parameters such as the distance and visibility in hazy images. Furthermore, the grey system theory allows for the recovery of clear images by applying a reverse operation. This methodology has found utility in scenarios involving adverse weather conditions, and it offers a valuable approach to mitigate the impact of fog and haze on image quality [188, 189].

6.3 Dehazing Model

This section provides an overview of the AGD-NET dehazing model's architecture. The model is structured as a symmetrical Encoder-Decoder network, integrating Dense Blocks, Inception Blocks, and an Attention Mechanism.

6.3.1 Architecture

The integration of dense connections [190], which concatenate the outputs of preceding layers in the architecture with the input of the current layer, substantially enhances convolutional neural networks. This design promotes feature reuse, fosters parameter efficiency, and effectively mitigates the vanishing gradient problem. Figure 6.2 represents the overall architecture of the AGD-NET model. The AGD-NET model architecture incorporates a Dense Block, consisting of four consecutive 3x3 Convolutional Blocks, as depicted in Figure 6.3. Each Convolutional Block takes as input the concatenated outputs of all prior blocks at the same spatial resolution. The output features of each Convolutional Block contain one-fourth the number of channels compared to the input feature map of the Dense Block.

The Attention mechanism [191] computes the similarity between any pair of values in a sequence or non-sequential data. In image data, self-attention reveals global relationships among pixels. In Figure 6.4, the input feature map under-

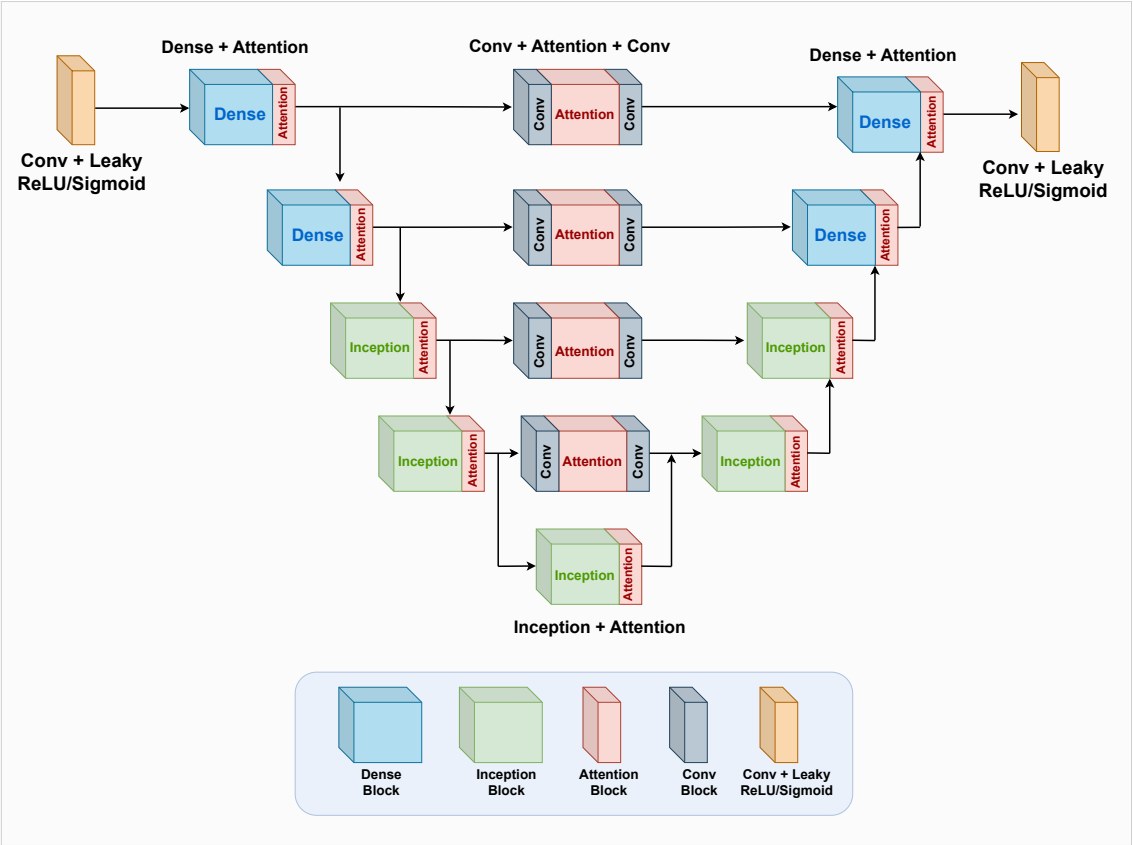


Figure 6.2: AGD-NET Architecture

goes three 1x1 Convolutional layers to generate the Query, Key, and Value feature maps. Matrix multiplication of the Query and Key maps yields attention scores, which are subsequently normalized using softmax to produce attention weights. The final output of the attention block is derived by adding the input feature map to the element-wise product of the attention weights and the Value feature map, utilizing a residual connection. This Attention Block is implemented for each network stage. To enhance network width and process input features at multiple scales in each stage, the Inception Network [192] is employed. Figure 6.5 illustrates the Inception Block, featuring four multi-scale Convolutional Blocks with 1x1, 3x3, 5x5, and 7x7 kernels, each contributing a quarter of the input channels. The outputs of these blocks are concatenated and passed to a final 3x3 Convolutional Block.

In the original U-Net architecture proposed by Ronneberger et al. [193], skip con-

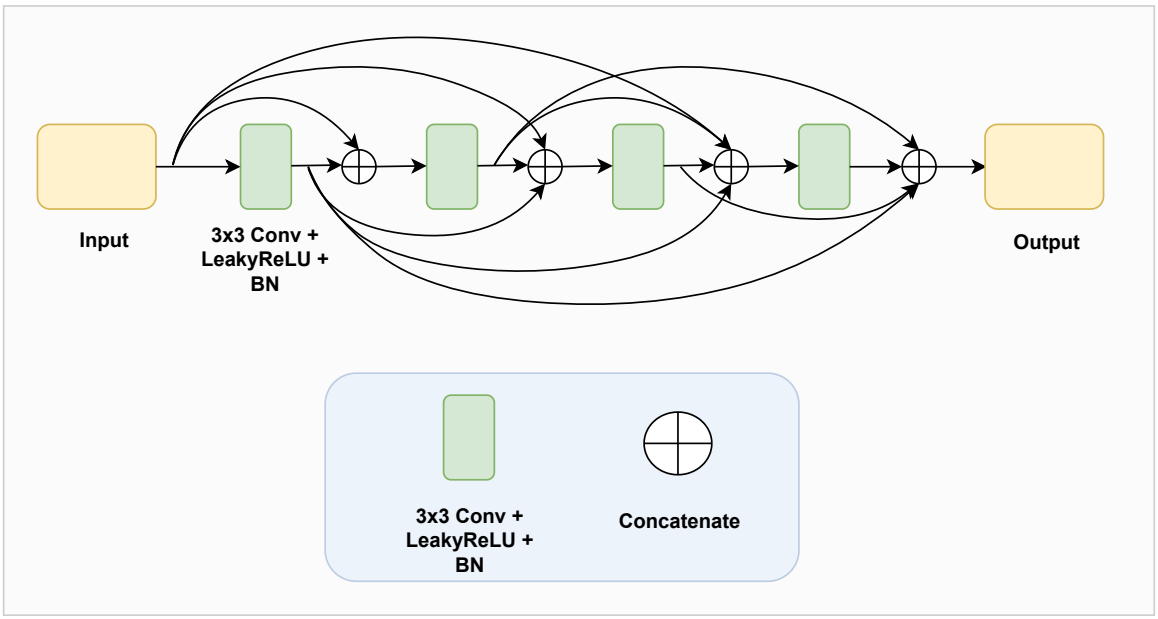


Figure 6.3: Dense Block

nections were employed for feature reuse and to provide context from encoder layers to the decoder layers. However, it is common to modify these skip connections to allow further processing of encoder feature maps before concatenation with the decoder layer input. This modification grants the model increased flexibility in representing the encoder feature maps before concatenation, and this processing can be accomplished through learnable layers. Consequently, the choice is made to preprocess the encoder feature maps with a sequence of a Convolutional Block, a Self-Attention Block, and another Convolutional Block before concatenating with the decoder layer output. LeakyReLU activation and Instance Normalization are applied after all learnable layers within the model.

6.3.2 Loss Functions

The AGD-NET dehazing model employs a combination of three distinct loss functions: L1 loss, SSIM (structural similarity index measure) loss, and perceptual content loss.

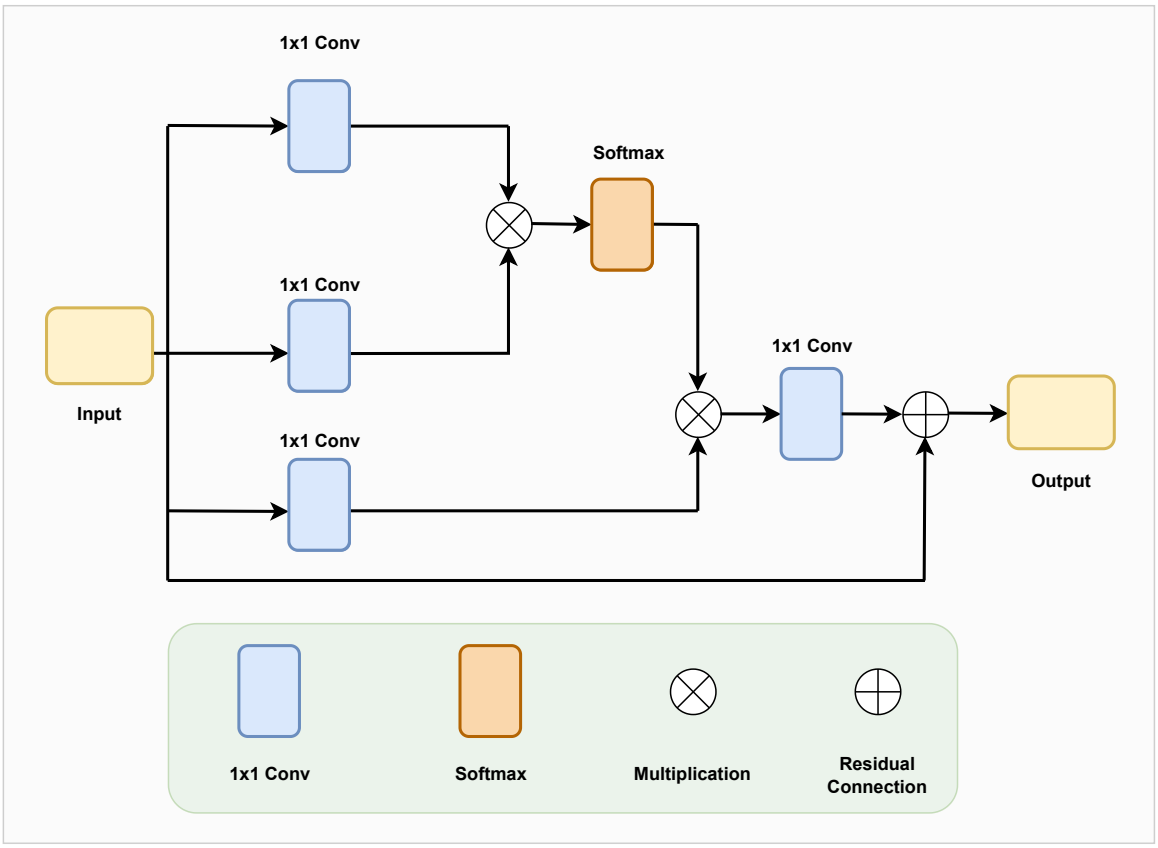


Figure 6.4: Attention Block

6.3.2.1 L1 Loss

The L1 loss computes the absolute difference on a per-pixel basis between the output and ground truth images. It is formulated as follows:

$$L_1 = \frac{1}{N}\sum_{i=1}^{N} |y_{i,true} - y_{i,predicted}|$$

Preference is given to the L1 loss over the L2 loss as the L2 loss tends to be more responsive to noise points due to the square factor. The L1 loss has shown to produce better results compared to the L2 loss in image restoration tasks [194]. To generate visually appealing outcomes, the model incorporates perceptually motivated loss functions, including the SSIM loss and the Perceptual loss.

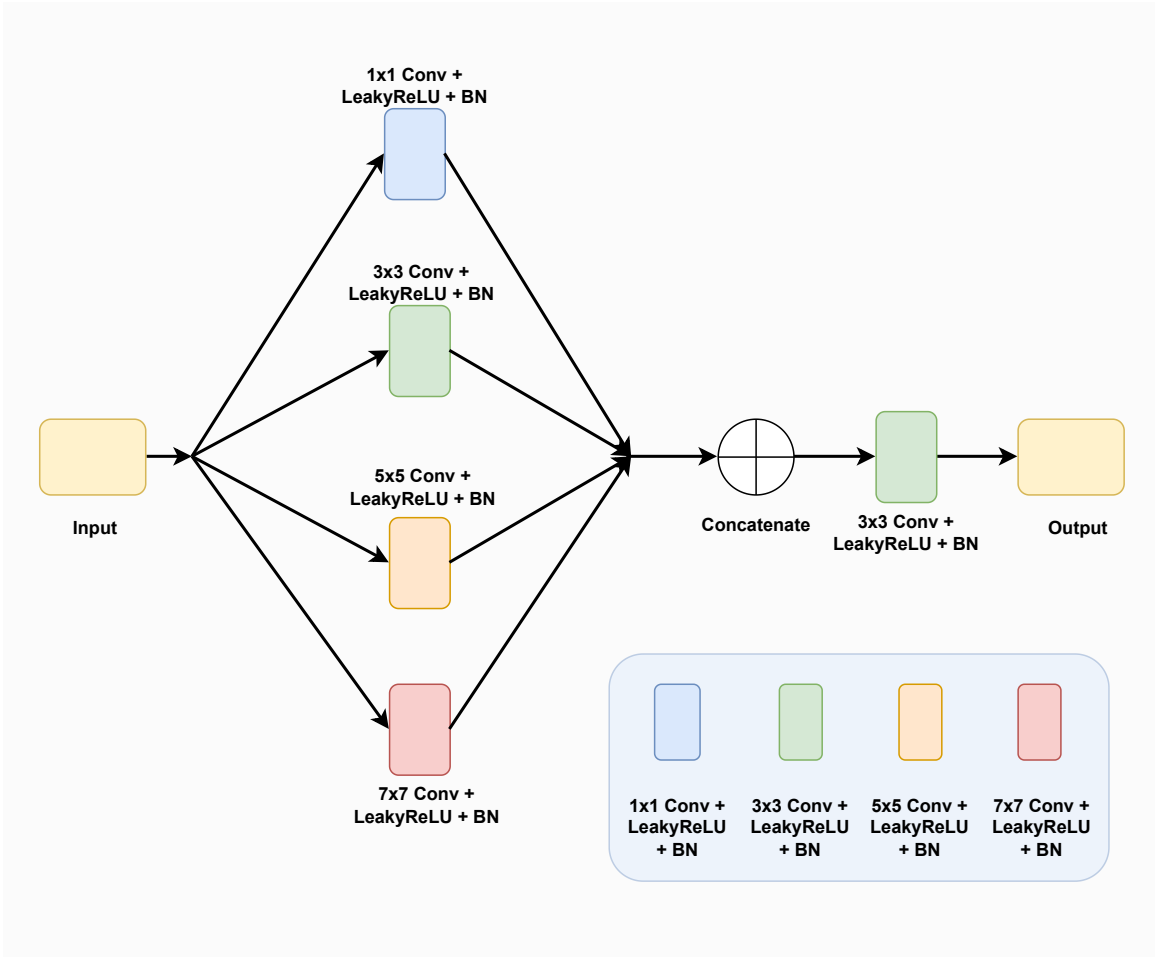


Figure 6.5: Inception Block

6.3.2.2 SSIM Loss

The SSIM (structural similarity index measure) metric [195] is designed to compare images by evaluating their luminance, contrast, and structural similarities. It is formulated as follows:

$$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_x^2 + C_1)(\sigma_x^2 + \sigma_x^2 + C_2)}$$

where, μ = Mean of all pixel values

σ = Standard Deviation of all pixel values

C_1, C_2 = Constants

The loss form of the metric is,

$$L_{SSIM} = 1 - SSIM$$

6.3.2.3 Perceptual Content Loss

The perceptual loss [196] is designed to compare images at a higher-level representation, in contrast to the lower-level pixel-wise comparison in the L1 loss. Higher-level features are extracted using a pre-trained network, and the per-pixel loss is computed for these extracted features as the content loss given below.

$$L_{content} = \frac{1}{N} \sum_{i=1}^{N} |\Phi(y_{true})_i - \Phi(y_{predicted})_i|$$

where, Φ = feature extractor

The VGG-16 pre-trained network has been utilized, 3^{rd}, 7^{th}, 10^{th} and 14^{th} layers of the network have been used.

Thus, the objective function for the entire network is:

$$L_{total} = \alpha * (L_1 + L_{SSIM}) + L_{content}$$

where, α = 10

6.4 Experiments

6.4.1 Hazy Image Dataset Construction

At present, only a limited number of datasets encompass realistic hazy images [5–8], while many others primarily consist of synthetic data generated using CARLA

Dataset	Type	I/ O	P/ N
Dense-Haze [5]	Realistic	Outdoor & Indoor	Pair
O-HAZE [6]	Realistic	Outdoor	Pair
I-HAZE [7]	Realistic	Indoor	Pair
NH-HAZE [8]	Realistic	Outdoor	Pair
RESIDE-β SOTS Outdoor [9]	Realistic	Outdoor	Pair

Table 6.1: Hazy image dataset details

[197] simulation [198–202]. Training models on such synthetic datasets can lead to a significant challenge wherein models may perform well with the dataset used for training but may struggle to dehaze real-world images or footage. Therefore, for the development of the AGD-NET model exclusively utilize datasets containing realistic hazy images to ensure its functionality in real-world scenarios. The O-HAZE (Outdoor Haze) [6], NH-HAZE (Non-Homogeneous Haze) [8], Dense-Haze (Dense Haze) [5], and I-HAZE (Indoor Haze) [7] datasets are employed for the training, validation, and testing of the AGD-NET model. These datasets consist of hazy images generated using a haze generator or machine. Table 6.1 provides an overview of the datasets used during model development, specifying the dataset type (Realistic or Synthetic (CARLA [197])), whether the images are indoor or outdoor, and whether they are paired or non-paired. In the assessment of AGD-NET's on-road dehazing capabilities for ADAS-enabled vehicles the RESIDE-β SOTS Outdoor [9] Dataset is utilized. To comprehensively evaluate the model's performance in real-world scenarios, AGD-NET is meticulously trained, validated, and tested on this dataset. By utilizing this robust dataset, the aim is to provide a thorough analysis of AGD-NET's effectiveness in enhancing visibility and ensuring optimal driving conditions for ADAS-equipped vehicles operating in outdoor environments. The utilization of RESIDE-β SOTS Outdoor Dataset allows to assess the model's adaptability and reliability in practical on-road settings, providing valuable insights for the integration of dehazing techniques in ADAS applications.

The integration of multiple datasets serves the purpose of enhancing the model's robustness and adaptability, with the aim of ensuring its effectiveness in real-world scenarios in the future.

6.4.2 Training Details

Given the limited size of the datasets, extensive data augmentation is employed to ensure effective learning for the model. The augmentation process involves generating 100 random crops for each hazy image every 25 epochs. The AGD-NET model undergoes training for a total of 100 epochs, resulting in four rounds of augmenting the dataset with random crops during the training process. Since the original dataset images possess high dimensions, this approach allows for the use of relatively large crop sizes, thereby improving feature extraction. Consequently, 512x512 sized image crops serve as input to the network. The training process employs the Adam optimizer with a learning rate of 1e-4, along with beta1 and beta2 values set to 0.9 and 0.999, respectively. The training is conducted on an Nvidia GeForce RTX 3090.

6.5 Evaluation and Discussion

6.5.1 Quantitative Evaluation of the Method

For the evaluation of dehazing algorithms, two key metrics were employed: Peak-Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). PSNR gauges the pixel-wise disparities between the original and dehazed images, while SSIM evaluates the structural similarities and perceptual aspects. The objective of dehazing algorithms is to enhance the clarity and sharpness of images captured in hazy or foggy conditions. PSNR helps assess their ability to mitigate haze effects, while SSIM assesses their capacity to preserve structural details

Models	**Dense Haze** [5]		**O-Haze** [6]		**I-Haze** [7]		**NH-Haze** [8]	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DCP [203]	10.43	0.39	13.28	0.52	12.88	0.62	12.93	0.48
AOD-Net [204]	10.66	0.37	18.57	0.68	15.40	0.72	13.48	0.45
EDN-GTM [205]	15.43	0.52	23.46	0.81	**22.90**	**0.82**	20.24	0.71
SRDefog [206]	16.67	0.5	**24.61**	0.75	-	-	**20.99**	0.61
AGD-Net(Ours)	**17.76**	**0.59**	23.57	**0.82**	19.74	0.81	19.30	**0.72**

Table 6.2: Model comparison on Dense Haze [5], O-Haze [6], I-Haze [7], and NH-Haze [8] dataset

and overall image quality during dehazing. SSIM holds particular significance as it incorporates factors such as luminance, contrast, and structure, which align closely with human visual perception. Consequently, utilizing both PSNR and SSIM metrics offers a more comprehensive evaluation of dehazing algorithms.

The AGD-NET model, along with several benchmarked models, was tested on various datasets using PSNR and SSIM. The performance of the AGD-NET model and previous dehazing models is summarized in Table 6.2, with a focus on PSNR and SSIM. As shown, the AGD-NET model achieves comparable performance to the current state-of-the-art approaches on these datasets and excels by establishing a new state-of-the-art on the Dense-Haze [5] dataset.

The evaluation of the proposed AGD-NET model across various datasets reveals noteworthy achievements in performance. Particularly, the AGD-NET model establishes a new benchmark on the Dense Haze [5] dataset, surpassing the performance of existing models. This accomplishment underscores the model's exceptional ability to handle dense haze scenarios.

When assessing the AGD-NET model on the O-haze [6] dataset, it excels in terms of the Structural Similarity Index (SSIM), outperforming other models. In terms of Peak Signal-to-Noise Ratio (PSNR), the AGD-NET model secures the second-best position, with SRDefog leading. This highlights the model's proficiency in generating images with high structural fidelity, as evidenced by superior SSIM

scores.

Similarly, on the NH-haze [8] dataset, the AGN-NET model achieves the highest SSIM values among all compared models, indicating that the structures of objects in the AGD-NET model's generated images closely resemble those in the reference (haze-free) images. In terms of PSNR, the AGD-NET model attains the second-best scores, with SRDefog obtaining the highest PSNR values.

Throughout the evaluation across these datasets, the AGD-NET model consistently outperforms other models in terms of SSIM, emphasizing its ability to preserve object structures and maintain fidelity to reference images. This consistency underscores the efficacy and reliability of the proposed AGD-NET model in addressing the challenges of image dehazing.

In the case of the I-haze [7] dataset, while the AGD-NET model does not achieve record-breaking results in terms of PSNR and SSIM metrics, it comes close to the current benchmark results for SSIM, approximately 0.81, and surpasses the performance of both DCP [203] and AOD-Net [204]. The distinction in environmental conditions is significant, as indoor scenes often contain more indoor objects and denser, closer fog, unlike outdoor scenes with more distant fog. This variation in fog properties and scene composition in indoor scenes presents unique challenges that may impact the AGD-NET model's performance on indoor scenes like I-haze [7].

Additionally, the performance of the AGD-NET model is assessed on the RESIDE-β SOTS Outdoor Dataset [9]. Comparative results of AGD-NET against other state-of-the-art models are presented in Table 6.3. Notably, AGD-NET demonstrates superior dehazing capabilities, outperforming all competing models. It achieves the highest Peak Signal-to-Noise Ratio (PSNR) with a value of 24.81 and a Structural Similarity Index (SSIM) score of 0.9424.

Models	**RESIDE-β SOTS Outdoor Dataset** [9]	
	PSNR ↑	**SSIM** ↑
DCP [112]	19.13	0.8148
GFN [111]	21.55	0.8444
DCPDN [110]	19.93	0.8449
DehazeNet [207]	22.46	0.8514
Enh. Pix2Pix [121]	22.57	0.8630
AOD-Net [204]	20.29	0.8765
MADN [208]	23.64	0.9137
AGD-NET (Ours)	**24.81**	**0.9424**

Table 6.3: Model comparison on RESIDE-β SOTS Outdoor Dataset [9, 10]

6.5.2 Qualitative Evaluation of the Method

A comprehensive qualitative evaluation is presented, demonstrating the effectiveness of the proposed AGD-NET model in dehazing images captured in hazy weather conditions. The model is visually compared with existing methods to showcase its superior performance. Experiments were conducted with the AGD-NET model, as well as other well-known models, including DCP [203], AOD-NET [204], EDN-GTM [205], and SRDefog [206], which have previously achieved benchmark results in dehazing tasks on various datasets, including Dense Haze [5], O-Haze [6], I-Haze [7], and NH-Haze [8]. The AGD-NET model demonstrates its ability to handle various scenarios and produce haze-free images with clear color and contrast details.

In Figure 6.6, which displays results on Dense Haze [5] images in its first row, the model excels in preserving image sharpness and structure during dehazing. However, there is a notable challenge in maintaining color contrast compared to the ground truth. In the second row of comparison images, it becomes evident that the model's dehazed images closely align with the ground truth, outperforming DCP [203], AOD-NET [204], EDN-GTM [205], and SRDefog [206]. Notably, DCP [203] and AOD-NET [204] produce darker and noisier results, while EDN-GTM [205] and SRDefog [206] yield blurry outputs.

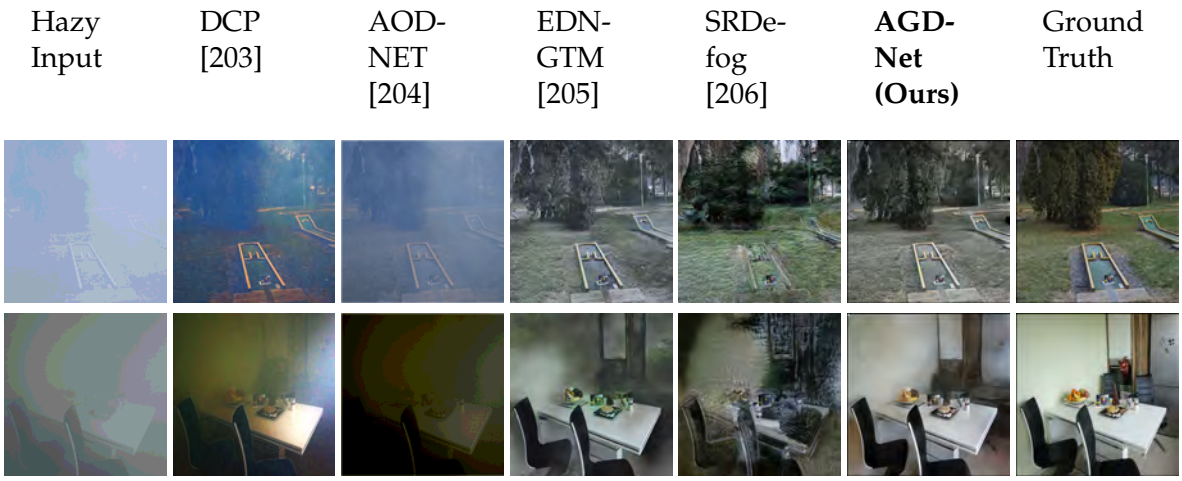


Figure 6.6: Visual comparison of the results on the Dense Haze [5] dataset

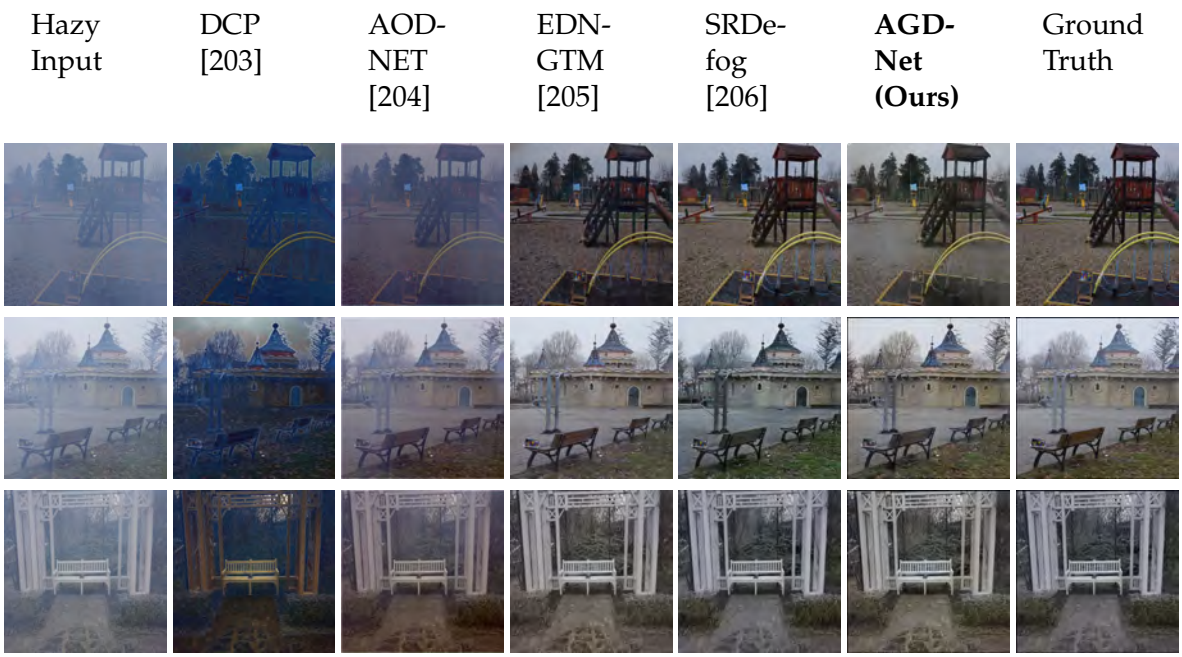


Figure 6.7: Visual comparison of the results on the O-Haze [6] dataset

Turning to Figure 6.7, which features O-haze dataset images, the first-row images show SRDefog [206] as the standout performer, although some residual haze remains in the generated images. In the second and third rows, the model excels, producing haze-free images while maintaining sharpness and object structures.

Figure 6.8, dedicated to I-haze [7] dataset images, indicates that the model faces challenges in achieving optimal results, although it surpasses DCP [203] and

Figure 6.8: Visual comparison of the results on the I-Haze [7] dataset

AOD-NET [204]. While not achieving top performance, it demonstrates an improvement over these counterparts.

In Figure 6.9, focusing on NH-haze [8] dataset images, the first and second rows highlight the model's proficiency in generating images that closely resemble the ground truth, preserving sharpness and structure. Conversely, SRDefog [206] excels in color contrast compared to the haze-free ground truth images.

Furthermore, a comprehensive assessment of the AGD-NET model's on-road dehazing capabilities is conducted using the RESIDE-β SOTS Outdoor Dataset [9]. The AGD-NET model is trained and evaluated using the RESIDE-β SOTS Outdoor Dataset. Figure 6.10 illustrates the hazy input image used for evaluation, the dehazed results generated by the AGD-NET model, and the ground truth image, which serves as the haze-free reference. This visual comparison provides a clear understanding of the dehazing performance. Notably, the dehazed images produced by AGD-NET exhibit remarkable clarity and are essentially free from

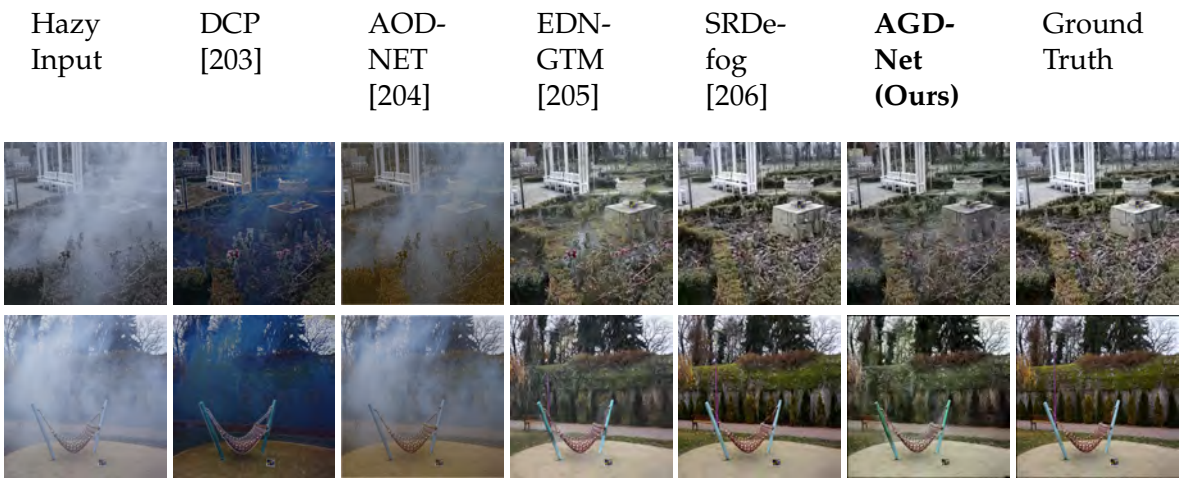


Figure 6.9: Visual comparison of the results on the NH-Haze [8] dataset

haze. However, it's worth mentioning that in a subset of images, there is a minor issue of atmospheric color shift, as exemplified in Row 2 of Figure 6.10. Despite this minor concern, AGD-NET demonstrates an exceptional overall dehazing performance. The comprehensive evaluation underscores the model's outstanding on-road dehazing capabilities, which are particularly advantageous for ADAS in ensuring clear and safe vehicle driving conditions.

6.5.3 Failure Case

While the proposed AGD-NET model consistently demonstrates remarkable performance across a wide array of images, it is important to acknowledge its limitations, as it may encounter challenges in specific scenarios. As illustrated in Figure 6.11, two cases are presented where the AGD-NET model faces difficulties and, regrettably, fails to preserve the original image structure during the dehazing process. In the first row, the image predominantly consists of extensive areas of green and white, a visual characteristic that poses a formidable challenge for the AGD-NET model. This unique configuration leads to instances where objects within the scene become intricately camouflaged, rendering them indistinguishable to our model's dehazing mechanism. Similarly, in the second row, the presence of a

Figure 6.10: Visual comparison of the AGD-NET model's generated results on the RESIDE-β SOTS Outdoor Dataset [9]

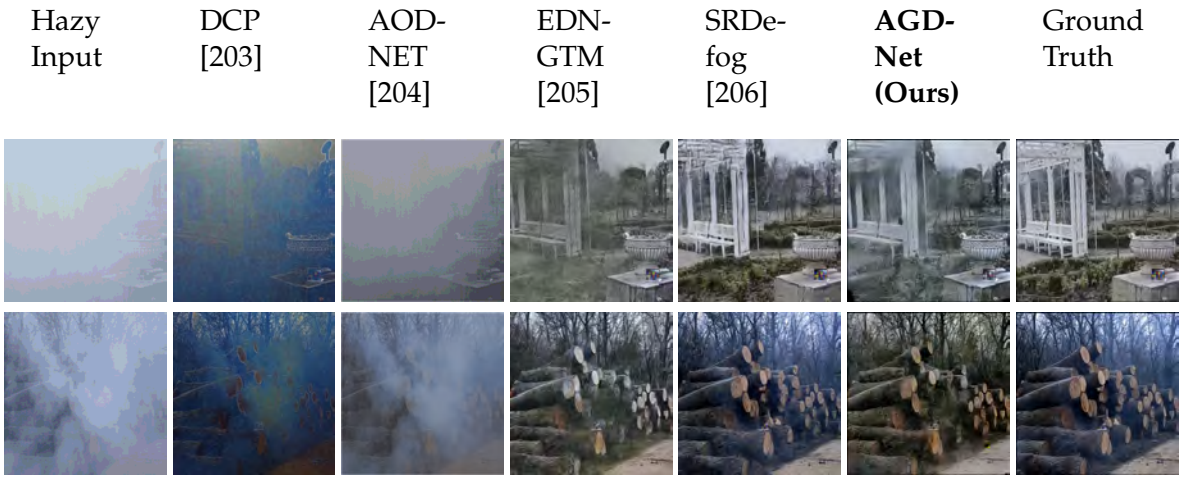


Figure 6.11: Some of the failure cases of the results generated by our model.

wood texture further compounds the complexity of the scene. The intricate texture patterns obscure the cylindrical shape of wooden blocks, thus confounding the AGD-NET model's ability to accurately discern and restore object structures. Consequently, portions of the output exhibit blurriness, particularly in the context of object structures and boundaries.

6.6 Challenges and Future Research Directions

In the current AGD-NET model version, the proposed model demonstrates significant advancements in image dehazing. However, certain limitations are acknowledged, which are intended to be addressed and overcome in future research endeavors. Specifically, the AGD-NET model encounters difficulties in scenarios where the haze is in close proximity to the camera and in areas where objects become camouflaged by the haze. For future research direction, a commitment is made to tackle these specific challenges and further refine the model's capabilities. The research agenda will include a dedicated focus on addressing scenarios with close-range haze and improving the AGD-NET model's ability to handle camouflaged areas effectively.

Additionally, the need to expand research to encompass indoor scenes is recognized, as they present their own unique set of challenges in the context of image dehazing. Indoor scenes often feature distinctive characteristics and complexities that require specialized approaches for haze removal. As part of future research, a dive into the intricacies of indoor haze scenarios is planned to develop tailored solutions that enhance the versatility and applicability of the AGD-NET model.

6.7 Summary

This chapter has introduced the innovative AGD-Net for single-image dehazing. The AGD-Net employs a U-Net style architecture with an Attention-Guided Dense Inception encoder-decoder framework for image dehazing. Unlike prior models that heavily rely on synthetic datasets, often derived from CARLA simulation, proposed approach distinguishes itself by being exclusively trained and evaluated on real-world data, thereby enhancing the effectiveness and reliability of the AGD-Net model in practical scenarios. Through a comprehensive experi-

mental analysis conducted on five benchmark haze datasets, it has been demonstrated that the AGD-Net network performs on par with existing state-of-the-art CNN and GAN-based methods, as assessed through both quantitative and qualitative evaluations. Moreover, visual comparisons with other dehazing techniques reaffirm the superior performance of AGD-Net, firmly establishing it as a leading solution in the realm of image dehazing. The proposed AGD-Net stands as a pioneering solution for on-road image dehazing in the context of ADAS-enabled vehicles. This innovation empowers other detection and recognition systems to function seamlessly even in hazy weather conditions, thereby contributing to the advancement of ADAS-enabled vehicles' capabilities and safety.

Chapter 7

Conclusion and Future Work

In conclusion, the integration of AI into ADAS-enabled vehicles represents a pivotal milestone in enhancing the overall performance, efficiency, safety, security, and comfort of these vehicles. This thesis has explored various dimensions of this technological advancement and its far-reaching implications for the automotive industry, especially for ADAS-enabled vehicles and road safety.

Chapter 3, titled "Security Provisioning in VANETs for ADAS-Equipped Vehicles," constitutes a comprehensive investigation into enhancing the security of ADAS-enabled vehicles within VANETs. This chapter presents three innovative models, each contributing to the safeguarding of the VANET environment by effectively detecting various types of attacks. The first model, the Multi-branch Reconstruction Error (MbRE) model, is a deep learning-based intrusion detection model designed with a specific focus on intelligent edge-based policing within VANETs. This model capitalizes on advanced deep learning techniques to identify and thwart security threats in real-time. The second model, known as HybridSecNet, introduces a two-step Intrusion Detection System (IDS) tailored explicitly for mitigating in-vehicle communication security threats. It offers enhanced protection, particularly in the realm of Controller Area Networks (CAN), where security

vulnerabilities can have critical implications. The third model, SCAN-GAN, addresses the generation of synthetic Controller Area Network (CAN) data, demonstrating the adaptability of Generative Adversarial Networks (GANs) in working with diverse data types. SCAN-GAN showcases the customizable nature of GANs, enabling the generation of data specific to the needs of the security system. The chapter systematically evaluates the intrusion detection capabilities of each model and conducts comparative analyses with existing models. These assessments conclusively demonstrate that the proposed models surpass their counterparts, setting a new benchmark in the domain of VANET and ADAS security. This chapter highlights the critical importance of advanced security provisions for ADAS-equipped vehicles in VANET environments, ensuring a safer and more secure driving experience.

Chapter 4 introduces a groundbreaking hardware pipeline and a specialized approach for real-time microsleep detection, specifically tailored for ADAS. The proposed methodology leverages the NeuroSky MindWave Mobile EEG device to capture individual brainwave signals, seamlessly transmitting this vital data to the Jetson Nano platform via Bluetooth technology. On the Jetson Nano platform, a well-established machine learning-driven microsleep detection model takes center stage. This model is enriched with advanced techniques, including Wavelet transform and Short Time Fourier Transform (STFT), which enhance its ability to classify input data into distinct states: "wakefulness," "Microsleep (MSE)," and "Deep Sleep." The system's performance stands out with its remarkable precision in real-time microsleep detection, ensuring timely and accurate identification of microsleep events. The results are readily accessible through an intuitive web interface, allowing for immediate intervention when necessary. The proposed research signifies a significant advancement in the realm of microsleep detection, a critical concern for ADAS. The rapid and accurate identification of microsleep events is of paramount importance, and the proposed approach promises to en-

hance the safety and reliability of ADAS by proactively addressing driver fatigue and drowsiness, ultimately contributing to safer road experiences for all.

Chapter 5 introduces an innovative framework designed to tackle traffic congestion management at intersections by harnessing the power of Federated Learning and Vertical partitioning. This proposed technique addresses the pressing issue of vehicles idling at red lights, with a primary objective of improving fuel consumption in Intelligent Transportation Systems (ITS), which encompass ADAS-enabled vehicles as well as other vehicles. The framework showcases remarkable efficiency in enhancing the passing-vehicle ratio at intersections, ultimately leading to more streamlined traffic flow. Through a mathematical approach, the chapter sheds light on the intricate relationship between fuel consumption and travel time. It underscores the pivotal role of reducing idling time in achieving fuel conservation and lower carbon emissions, a significant concern in modern transportation. The evaluation of the proposed model is carried out using various parameters, confirming its effectiveness. Notably, the passing-vehicle ratio achieves an impressive 1.33, surpassing the traditional model's ratio of 0.88. These findings highlight the substantial advantages of the proposed framework over conventional vehicle passing per unit time ratio approaches, making it a promising and innovative solution for intelligent traffic management. This chapter underscores the potential for transformative advancements in traffic control and fuel efficiency in ADAS and ITS.

Chapter 6 presents an innovative approach for enhancing on-road perception for ADAS-enabled vehicles, particularly in adverse weather conditions. The centrepiece of this chapter is the AGD-NET model, which embodies an Attention-Guided Dense Inception U-Net architecture designed to dehaze hazy images and deliver pristine, haze-free images as output. The evaluation of AGD-NET is both quantitative and qualitative, comparing its performance with state-of-the-art mod-

els in the field. The results of these comparisons unequivocally demonstrate that AGD-NET outperforms other models, setting a new benchmark in terms of critical metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Notably, AGD-NET excels in dehazing both general hazy scenarios and the specific challenges posed by on-road hazy conditions. This remarkable dehazing capability enhances the on-road perception of ADAS-enabled vehicles, ensuring that the vehicle's features operate seamlessly even in adverse weather conditions. The chapter underlines the significance of AGD-NET as a groundbreaking solution for enhancing the safety and reliability of ADAS in challenging environmental conditions.

In conclusion, the adoption of AI in ADAS-enabled vehicles represents a transformative step towards safer, more efficient, and comfortable transportation. The innovations presented in this thesis contribute to the advancement of ADAS technology, with a focus on enhancing security, perception, driver safety, and traffic management. These advancements have the potential to shape the future of ADAS and intelligent transportation systems, ultimately benefiting both individual vehicle occupants and society as a whole.

7.0.1 Future Work

The existing Intrusion Detection System operates efficiently under current conditions; however, with the network's expansion due to an increase in connected devices and advancing technology, vulnerabilities and potential attacks are likely to rise. To address this, it is imperative that the model undergoes continual updates through a process called Continual Learning Implementation. This approach involves developing mechanisms for ongoing learning, allowing the model to adapt to evolving network environments and scenarios by consistently updating itself with new traffic logs and trends.

In parallel, ongoing research aims to detect microsleep among drivers, a condition exacerbated by today's increasing vehicle numbers and risky driving behaviors. There is a critical need for further implementation of behavior and situation detection systems that utilize the same hardware to continuously monitor driver behavior. Such systems should not only assess but also respond to emergencies by alerting other road users, thereby enhancing passenger and vehicular safety.

Regarding traffic monitoring, the proposed mechanisms perform adequately under normal conditions, but their effectiveness during peak traffic hours, when vehicle flow significantly increases, remains uncertain. To manage and optimize traffic flow during these critical periods, the utilization of drones could be instrumental. These drones could dynamically adjust traffic signals and patterns to enhance transportation efficiency. Additionally, the current traffic management systems lack specialized configurations for emergency situations. Implementing an attention mechanism that prioritizes vehicles like ambulances or those carrying VIPs could ensure these vehicles receive the right of way by clearing their paths promptly, thus improving response times and overall traffic management during emergencies.

While ADAS-enabled vehicles have demonstrated various benefits compared to traditional vehicles, there remain challenges that necessitate further research and innovation to overcome [209]. Despite the development of computer vision algorithms capable of functioning in adverse weather conditions and enhancing camera capabilities in ADAS, certain outcomes may compromise image parameters such as colors or edges. Moreover, current computer vision algorithms, encompassing object detection, segmentation, style transfer, and image enhancement, face difficulties in scenarios involving camouflage, commonly encountered in everyday scenarios like pedestrian clothing or vehicle colour.

Another significant open challenge in ADAS-enabled vehicles arises during over-

taking manoeuvres [210]. Overtaking involves a complex set of actions, including lane changes, acceleration, de-acceleration, and other activities controlled by ADAS [210]. Ensuring the optimal performance of these ADAS operations during overtaking is crucial, as malfunctions in these operations can lead to accidents. Addressing this challenge is paramount for enhancing safety and efficiency in transportation facilitated by ADAS [210]. Efforts in ongoing research and innovation aimed at addressing these challenges will contribute to a more robust and reliable ADAS system. Overcoming these obstacles is essential for ensuring the continued advancement of ADAS technology and its role in providing a safer and more efficient mode of transportation.

References

[1] C. Rödel, S. Stadler, A. Meschtscherjakov, and M. Tscheligi, "Towards autonomous cars: The effect of autonomy levels on acceptance and user experience," in *Proceedings of the 6th international conference on automotive user interfaces and interactive vehicular applications*, 2014, pp. 1–8.

[2] K. Doghramji, M. M. Mitler, R. B. Sangal, C. Shapiro, S. Taylor, J. Walsleben, C. Belisle, M. K. Erman, R. Hayduk, R. Hosn *et al.*, "A normative study of the maintenance of wakefulness test (mwt)," *Electroencephalography and clinical neurophysiology*, vol. 103, no. 5, pp. 554–562, 1997.

[3] H.-G. Anneke, S. Jelena, M. Alexander, A. Peter, M. Johannes, and S. D. R., "Maintenance of wakefulness test (mwt) recordings," Sept. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3251716

[4] A. Malafeev, A. Hertig-Godeschalk, D. Schreier, J. Skorucak, J. Mathis, and P. Achermann, "Automatic detection of microsleep episodes with deep learning," 09 2020.

[5] C. O. Ancuti, C. Ancuti, M. Sbert, and R. Timofte, "Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images," in *2019 IEEE international conference on image processing (ICIP)*. IEEE, 2019, pp. 1014–1018.

[6] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer, "O-haze: a dehazing benchmark with real hazy and haze-free outdoor images," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 754–762.

[7] C. Ancuti, C. O. Ancuti, R. Timofte, and C. De Vleeschouwer, "I-haze: a dehazing benchmark with real hazy and haze-free indoor images," in *Advanced Concepts for Intelligent Vision Systems: 19th International Conference,*

ACIVS 2018, Poitiers, France, September 24–27, 2018, Proceedings 19. Springer, 2018, pp. 620–631.

[8] C. O. Ancuti, C. Ancuti, and R. Timofte, "Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 444–445.

[9] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2019.

[10] A. Mehra, M. Mandal, P. Narang, and V. Chamola, "Reviewnet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4256–4266, 2020.

[11] S. Research. (2022) Advanced driver assistance systems market. [Online]. Available: https://www.iot-now.com/2022/11/03/125096-advanced-driver-assistance-systems-market-to-exceed-us75bn-globally-by-2030-says-sm-research/

[12] Y. Xu, Z. Ye, and C. Wang, "Modeling commercial vehicle drivers' acceptance of advanced driving assistance system (adas)," *Journal of intelligent and connected vehicles*, vol. 4, no. 3, pp. 125–135, 2021.

[13] C. Gutierrez, M. Juliato, S. Ahmed, and M. Sastry, "Detecting attacks against safety-critical adas based on in-vehicle network message patterns," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Industry Track*. IEEE, 2019, pp. 9–12.

[14] M. Mishra and A. Kumar, "Adas technology: a review on challenges, legal risk mitigation and solutions," *Autonomous Driving and Advanced Driver-Assistance Systems (ADAS)*, pp. 401–408, 2021.

[15] A. Kurbanov, S. Grebennikov, S. Gafurov, and A. Klimchik, "Vulnerabilities in the vehicle's electronic network equipped with adas system," in *2019 3rd School on Dynamics of Complex Networks and their Application in Intellectual Robotics (DCNAIR)*. IEEE, 2019, pp. 100–102.

[16] X. Zhou, A. Schmedding, H. Ren, L. Yang, P. Schowitz, E. Smirni, and H. Alemzadeh, "Strategic safety-critical attacks against an advanced driver

assistance system," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2022, pp. 79–87.

[17] J. A. Onieva, R. Rios, R. Roman, and J. Lopez, "Edge-assisted vehicular networks security," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8038–8045, 2019.

[18] G. Bosurgi, O. Pellegrino, A. Ruggeri, and G. Sollazzo, "The effects of adas on driving behavior: a case study," *Sensors*, vol. 23, no. 4, p. 1758, 2023.

[19] M. A. Rahman and M. Mekker, "Common adas myths assumed by drivers and perpetuated by vehicle advertising," in *International Conference on Transportation and Development 2023*, 2023, pp. 194–206.

[20] E. Purwanto, D. Suleman, R. Sjarief, D. Sri, M. D. Juliansyah, and M. I. R. Trisandri, "Traffic accident prevention through acceptance of advanced driver assistance system (adas) among urban people." *Journal Européen des Systèmes Automatisés*, vol. 56, no. 3, 2023.

[21] M. Howard, G. Kennedy, and R. Pierce, "Microsleep literature review," *University of Melbourne*, 2001.

[22] J. Horne and J. Rumbold, "Sleep-related road collisions," *Medicine, Science and the Law*, vol. 55, no. 3, pp. 183–185, 2015.

[23] M. Q. Khan and S. Lee, "Gaze and eye tracking: Techniques and applications in adas," *Sensors*, vol. 19, no. 24, p. 5540, 2019.

[24] J. Yu, L. Wang, and X. Gong, "Study on the status evaluation of urban road intersections traffic congestion base on ahp-topsis modal," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 609–616, 2013.

[25] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[26] M. A. Taylor, J. E. Woolley, and R. Zito, "Integration of the global positioning system and geographical information systems for traffic congestion studies," *Transportation Research Part C: Emerging Technologies*, vol. 8, no. 1-6, pp. 257–285, 2000.

[27] P. Bou Samra, P. El Tomb, M. Hosni, A. Kassem, R. Rizk, S. Shayya, and S. Assaad, "Traffic congestion and blood pressure elevation: A comparative

cross-sectional study in lebanon," *The journal of clinical hypertension*, vol. 19, no. 12, pp. 1366–1371, 2017.

[28] Y. Zhao and Z. Tian, "An overview of the usage of adaptive signal control system in the united states of america," *Applied Mechanics and Materials*, vol. 178, pp. 2591–2598, 2012.

[29] J. Nidamanuri, C. Nibhanupudi, R. Assfalg, and H. Venkataraman, "A progressive review: Emerging technologies for adas driven solutions," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 326–341, 2021.

[30] L. Gauerhof, A. Kürzl, and M. Lienkamp, "Adas for the communication between automated and manually driven cars," in *7. Tagung Fahrerassistenzsysteme*, 2015.

[31] G. Qian, T. Fu, and L. Sun, "Research on the fuel consumption conservation potential of adas on passenger cars," in *E3S Web of Conferences*, vol. 268. EDP Sciences, 2021, p. 01035.

[32] A. S. Mohammed, A. Amamou, F. K. Ayevide, S. Kelouwani, K. Agbossou, and N. Zioui, "The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review," *Sensors*, vol. 20, no. 22, p. 6532, 2020.

[33] W. Y. Pao, L. Li, M. Agelin-Chaab, and J. Komar, "Drive-thru climate tunnel: A proposed method to study adas performance in adverse weather," SAE Technical Paper, Tech. Rep., 2023.

[34] T. Rahman, A. Liu, D. Cheema, V. Chirila, and D. Charlebois, "Adas reliability against weather conditions: Quantification of performance robustness," in *27th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration*, no. 23-0306, 2023.

[35] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: How rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE vehicular technology magazine*, vol. 14, no. 2, pp. 103–111, 2019.

[36] V. S. Saravanarajan, R.-C. Chen, C.-H. Hsieh, and L.-S. Chen, "Improving semantic segmentation under hazy weather for autonomous vehicles using explainable artificial intelligence and adaptive dehazing approach," *IEEE Access*, vol. 11, pp. 38 194–38 207, 2023.

[37] J. Li, R. Xu, J. Ma, Q. Zou, J. Ma, and H. Yu, "Domain adaptive object detection for autonomous driving under foggy weather," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 612–622.

[38] N. Sharma, N. Chauhan, and N. Chand, "Security challenges in internet of vehicles (iov) environment," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018, pp. 203–207.

[39] A. Samad, S. Alam, S. Mohammed, and M. Bhukhari, "Internet of vehicles (iov) requirements, attacks and countermeasures," in *Proceedings of 12th INDIACom; INDIACom-2018; 5th international conference on "computing for sustainable global development" IEEE conference, New Delhi*, 2018, pp. 1–4.

[40] Y. Sun, L. Wu, S. Wu, S. Li, T. Zhang, L. Zhang, J. Xu, and Y. Xiong, "Security and privacy in the internet of vehicles," in *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*. IEEE, 2015, pp. 116–121.

[41] S. K. Khan, N. Shiwakoti, P. Stasinopoulos, and Y. Chen, "Cyber-attacks in the next-generation cars, mitigation techniques, anticipated readiness and future directions," *Accident Analysis & Prevention*, vol. 148, p. 105837, 2020.

[42] S. G. Philipsen, B. Andersen, and B. Singh, "Threats and attacks to modern vehicles," in *2021 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*. IEEE, 2021, pp. 22–27.

[43] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis, "Intrusion detection systems for intra-vehicle networks: A review," *IEEE Access*, vol. 7, pp. 21 266–21 289, 2019.

[44] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for vanets: A statistical approach to rogue node detection," *IEEE transactions on vehicular technology*, vol. 65, no. 8, pp. 6703–6714, 2015.

[45] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and federated learning for collaborative intrusion detection in vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6073–6084, 2021.

[46] Y. Gao, H. Wu, B. Song, Y. Jin, X. Luo, and X. Zeng, "A distributed net-

work intrusion detection system for distributed denial of service attacks in vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 154 560–154 571, 2019.

[47] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "Securing the internet of vehicles: A deep learning-based classification framework," *IEEE networking letters*, vol. 3, no. 2, pp. 94–97, 2021.

[48] Q. Zhang, L. T. Yang, Z. Chen, and P. Li, "A survey on deep learning for big data," *Information Fusion*, vol. 42, pp. 146–157, 2018.

[49] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, 2017.

[50] Y. Dai, D. Xu, S. Maharjan, G. Qiao, and Y. Zhang, "Artificial intelligence empowered edge computing and caching for internet of vehicles," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 12–18, 2019.

[51] Z. Ning, P. Dong, X. Wang, L. Guo, J. J. Rodrigues, X. Kong, J. Huang, and R. Y. Kwok, "Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1060–1072, 2019.

[52] T. Alladi, V. Kohli, V. Chamola, F. R. Yu, and M. Guizani, "Artificial intelligence (ai)-empowered intrusion detection architecture for the internet of vehicles," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 144–149, 2021.

[53] L. Yang, A. Moubayed, and A. Shami, "Mth-ids: A multitiered hybrid intrusion detection system for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2021.

[54] P. F. de Araujo-Filho, G. Kaddoum, D. R. Campelo, A. G. Santos, D. Macêdo, and C. Zanchettin, "Intrusion detection for cyber–physical systems using generative adversarial networks in fog environment," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6247–6256, 2020.

[55] K. Agrawal, T. Alladi, A. Agrawal, V. Chamola, and A. Benslimane, "Novelads: A novel anomaly detection system for intra-vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[56] M. Adhikari, A. Munusamy, A. Hazra, V. G. Menon, V. Anavangot, and D. Puthal, "Security in edge-centric intelligent internet of vehicles: Issues

and remedies," *IEEE Consumer Electronics Magazine*, vol. 11, no. 6, pp. 24–31, 2021.

[57] A. Rajnák, "Lin-protocol, development tools, and software interfaces for local interconnect networks in vehicles."

[58] P. Milbredt, B. Vermeulen, G. Tabanoglu, and M. Lukasiewycz, "Switched flexray: Increasing the effective bandwidth and safety of flexray networks," in *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*. IEEE, 2010, pp. 1–8.

[59] H. K. Kim, "Toward automotive ethernet security," *Review of KIISC*, vol. 29, no. 1, pp. 34–37, 2019.

[60] B. Carnevale, L. Fanucci, S. Bisase, and H. Hunjan, "Macsec-based security for automotive ethernet backbones," *Journal of Circuits, Systems and Computers*, vol. 27, no. 05, p. 1850082, 2018.

[61] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 International Conference on Information Networking (ICOIN)*, 2016, pp. 63–68.

[62] S. Tariq, S. Lee, and S. S. Woo, "Cantransfer: Transfer learning based intrusion detection on a controller area network using convolutional lstm network," in *Proceedings of the 35th annual ACM symposium on applied computing*, 2020, pp. 1048–1055.

[63] R. S. Rathore, C. Hewage, O. Kaiwartya, and J. Lloret, "In-vehicle communication cyber security: challenges and solutions," *Sensors*, vol. 22, no. 17, p. 6679, 2022.

[64] A. Martínez-Cruz, K. A. Ramírez-Gutiérrez, C. Feregrino-Uribe, and A. Morales-Reyes, "Security on in-vehicle communication protocols: Issues, challenges, and future research directions," *Computer Communications*, vol. 180, pp. 1–20, 2021.

[65] G. Kaur and D. Kakkar, "Hybrid optimization enabled trust-based secure routing with deep learning-based attack detection in vanet," *Ad Hoc Networks*, vol. 136, p. 102961, 2022.

[66] V. Belenko, V. Chernenko, M. Kalinin, and V. Krundyshev, "Evaluation of gan applicability for intrusion detection in self-organizing networks of cy-

ber physical systems," in *2018 International Russian Automation Conference (RusAutoCon)*. IEEE, 2018, pp. 1–7.

[67] V. Sreerag, S. Aswin, A. A. Menon, and L. V. Namboothiri, "Reinforce nids using gan to detect u2r and r2l attacks," in *Ubiquitous Intelligent Systems: Proceedings of ICUIS 2021*. Springer, 2022, pp. 357–369.

[68] B. Tang, Y. Lu, Q. Li, Y. Bai, J. Yu, and X. Yu, "A diffusion model based on network intrusion detection method for industrial cyber-physical systems," *Sensors*, vol. 23, no. 3, p. 1141, 2023.

[69] Y. Wang, J. Ding, X. He, Q. Wei, S. Yuan, and J. Zhang, "Intrusion detection method based on denoising diffusion probabilistic models for uav networks," *Mobile Networks and Applications*, pp. 1–10, 2023.

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

[71] H. Kang, B. I. Kwak, Y. H. Lee, H. Lee, H. Lee, and H. K. Kim, "Car hacking: Attack & defense challenge 2020 dataset," 2021. [Online]. Available: https://dx.doi.org/10.21227/qvr7-n418

[72] F. Gonçalves, B. Ribeiro, O. Gama, J. Santos, A. Costa, B. Dias, M. J. Nicolau, J. Macedo, and A. Santos, "V2x security threats," Dec. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4304411

[73] J. Raiyn, "Data and cyber security in autonomous vehicle networks," *Transport and Telecommunication*, vol. 19, no. 4, pp. 325–334, 2018.

[74] T. Wang, Z. Cao, S. Wang, J. Wang, L. Qi, A. Liu, M. Xie, and X. Li, "Privacy-enhanced data collection based on deep learning for internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6663–6672, 2019.

[75] A. Neri, S. Baldoni, and R. Capua, "On cyber-security of augmentation networks," in *Proceedings of the 2019 International Technical Meeting of The Institute of Navigation*, 2019, pp. 408–422.

[76] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[77] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[78] S. Cloudin and P. M. Kumar, "Review of driver behavior detection methodologies in vanet," *International Journal of Pure and Applied Mathematics*, vol. 120, no. 6, pp. 1713–1728, 2018.

[79] H. Abbood, W. Al-Nuaimy, A. Al-Ataby, S. A. Salem, and H. S. AlZubi, "Prediction of driver fatigue: Approaches and open challenges," in *2014 14th UK Workshop on Computational Intelligence (UKCI)*. IEEE, 2014, pp. 1–6.

[80] M. Rivera, "Monitoring of micro-sleep and sleepiness for the drivers using eeg signal," 2013.

[81] B. Shang, C. Zhang, Y. Liu, L. Wang, and L. Li, "Predicting the association between fine-grained attributes and black-boxed perceptual performance," in *2020 35th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2020, pp. 661–666.

[82] M. M. Littlefield, *Instrumental intimacy: EEG wearables and neuroscientific control*. JHU Press, 2018.

[83] R. Roach, "Epilepsy, digital technology and the black-boxed self," *New Media & Society*, vol. 20, no. 8, pp. 2880–2897, 2018.

[84] Z. Liang and B. Ploderer, "How does fitbit measure brainwaves: a qualitative study into the credibility of sleep-tracking technologies," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 1, pp. 1–29, 2020.

[85] A. Balaji, U. Tripathi, V. Chamola, A. Benslimane, and M. Guizani, "Toward safer vehicular transit: Implementing deep learning on single channel eeg systems for microsleep detection," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[86] H. Phan, K. Mikkelsen, O. Y. Chén, P. Koch, A. Mertins, P. Kidmose, and M. De Vos, "Personalized automatic sleep staging with single-night data: a pilot study with kullback–leibler divergence regularization," *Physiological measurement*, vol. 41, no. 6, p. 064004, 2020.

[87] H. Seo, S. Back, S. Lee, D. Park, T. Kim, and K. Lee, "Intra-and inter-epoch temporal context network (iitnet) using sub-epoch features for automatic

sleep scoring on raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 61, p. 102037, 2020.

[88] A. Saeed, F. D. Salim, T. Ozcelebi, and J. Lukkien, "Federated self-supervised learning of multisensor representations for embedded intelligence," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1030–1040, 2020.

[89] H. Phan, O. Y. Chén, P. Koch, Z. Lu, I. McLoughlin, A. Mertins, and M. De Vos, "Towards more accurate automatic sleep staging via deep transfer learning," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 6, pp. 1787–1798, 2020.

[90] X. Jiang, "Mrnet: a multi-scale residual network for eeg-based sleep staging," *arXiv preprint arXiv:2101.02538*, 2021.

[91] H. Phan, O. Y. Chén, M. C. Tran, P. Koch, A. Mertins, and M. De Vos, "Xsleepnet: Multi-view sequential model for automatic sleep staging," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[92] E. Eldele, Z. Chen, C. Liu, M. Wu, C.-K. Kwoh, X. Li, and C. Guan, "An attention-based deep learning approach for sleep stage classification with single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 809–818, 2021.

[93] H. Prendinger, K. Gajananan, A. B. Zaki, A. Fares, R. Molenaar, D. Urbano, H. van Lint, and W. Gomaa, "Tokyo virtual living lab: Designing smart cities based on the 3d internet," *IEEE Internet Computing*, vol. 17, no. 6, pp. 30–38, 2013.

[94] W.-H. Lee and C.-Y. Chiu, "Design and implementation of a smart traffic signal control system for smart city applications," *Sensors*, vol. 20, no. 2, p. 508, 2020.

[95] W. Cao, Y. Wu, C. Chakraborty, D. Li, L. Zhao, and S. K. Ghosh, "Sustainable and transferable traffic sign recognition for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[96] J. Chen, Q. Wang, H. H. Cheng, W. Peng, and W. Xu, "A review of vision-based traffic semantic understanding in itss," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[97] M. Tsukada, T. Oi, M. Kitazawa, and H. Esaki, "Networked roadside perception units for autonomous driving," *Sensors*, vol. 20, no. 18, p. 5320, 2020.

[98] A. Jandial, P. Merdrignac, O. Shagdar, and L. Fevrier, "Implementation and evaluation of intelligent roadside infrastructure for automated vehicle with i2v communication," in *Vehicular Ad-hoc Networks for Smart Cities: Third International Workshop, 2019*. Springer, 2020, pp. 3–18.

[99] M. Waqas, S. Tu, S. U. Rehman, Z. Halim, S. Anwar, G. Abbas, Z. H. Abbas, and O. U. Rehman, "Authentication of vehicles and road side units in intelligent transportation system." *Computers, Materials & Continua*, vol. 64, no. 1, 2020.

[100] S. Tak, J. Yoon, S. Woo, and H. Yeo, "Sectional information-based collision warning system using roadside unit aggregated connected-vehicle information for a cooperative intelligent transport system," *Journal of advanced transportation*, vol. 2020, pp. 1–12, 2020.

[101] M. Massar, I. Reza, S. M. Rahman, S. M. H. Abdullah, A. Jamal, and F. S. Al-Ismail, "Impacts of autonomous vehicles on greenhouse gas emissions—positive or negative?" *International Journal of Environmental Research and Public Health*, vol. 18, no. 11, p. 5567, 2021.

[102] H. Rakha and R. K. Kamalanathsharma, "Eco-driving at signalized intersections using v2i communication," in *2011 14th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2011, pp. 341–346.

[103] M. Al-Turki, A. Jamal, H. M. Al-Ahmadi, M. A. Al-Sughaiyer, and M. Zahid, "On the potential impacts of smart traffic control for delay, fuel energy consumption, and emissions: An nsga-ii-based optimization case study from dhahran, saudi arabia," *Sustainability*, vol. 12, no. 18, p. 7394, 2020.

[104] C. A. Zimmerman, J. Marks, J. H. Jenq, C. Cluett, A. DeBlasio, J. Lappin, H. A. Rakha, K. Wunderlich *et al.*, "Phoenix metropolitan model deployment initiative evaluation report," 2000.

[105] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[106] Z. Du, B. H. Chaudhuri, and P. Pisu, "Distributed coordination of connected and automated vehicles at multiple interconnected intersections," *International Journal of Computer and Information Engineering*, vol. 10, no. 6, pp. 1015–1021, 2016.

[107] M. A. S. Kamal, J.-i. Imura, A. Ohata, T. Hayakawa, and K. Aihara, "Coordination of automated vehicles at a traffic-lightless intersection," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*. Ieee, 2013, pp. 922–927.

[108] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[109] A. Wang, W. Wang, J. Liu, and N. Gu, "Aipnet: Image-to-image single image dehazing with atmospheric illumination prior," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 381–393, 2018.

[110] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3194–3203.

[111] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang, "Gated fusion network for single image dehazing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3253–3261.

[112] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[113] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 154–169.

[114] H. Dong, J. Pan, L. Xiang, Z. Hu, X. Zhang, F. Wang, and M.-H. Yang, "Multi-scale boosted dehazing network with dense feature fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2157–2167.

[115] X. Liu, Y. Ma, Z. Shi, and J. Chen, "Griddehazenet: Attention-based multi-scale network for image dehazing," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7314–7323.

[116] H. Zhang, V. Sindagi, and V. M. Patel, "Multi-scale single image dehazing

using perceptual pyramid deep network," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 902–911.

[117] Q. Yi, J. Li, F. Fang, A. Jiang, and G. Zhang, "Efficient and accurate multi-scale topological network for single image dehazing," *IEEE Transactions on Multimedia*, vol. 24, pp. 3114–3128, 2021.

[118] X. Lian, Y. Pang, Y. He, X. Li, and A. Yang, "Learning tone mapping function for dehazing," *Cognitive Computation*, vol. 9, pp. 95–114, 2017.

[119] A. Mehta, H. Sinha, P. Narang, and M. Mandal, "Hidegan: A hyperspectral-guided image dehazing gan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 212–213.

[120] M. Fu, H. Liu, Y. Yu, J. Chen, and K. Wang, "Dw-gan: A discrete wavelet transform gan for nonhomogeneous dehazing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 203–212.

[121] Y. Qu, Y. Chen, J. Huang, and Y. Xie, "Enhanced pix2pix dehazing network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8160–8168.

[122] P. Wang, H. Zhu, H. Huang, H. Zhang, and N. Wang, "Tms-gan: A twofold multi-scale generative adversarial network for single image dehazing," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 5, pp. 2760–2772, 2021.

[123] Y. Dong, Y. Liu, H. Zhang, S. Chen, and Y. Qiao, "Fd-gan: Generative adversarial networks with fusion-discriminator for single image dehazing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 10 729–10 736.

[124] H. Zhu, X. Peng, V. Chandrasekhar, L. Li, and J.-H. Lim, "Dehazegan: When image dehazing meets differential programming." in *IJCAI*, 2018, pp. 1234–1240.

[125] D. Engin, A. Genç, and H. Kemal Ekenel, "Cycle-dehaze: Enhanced cyclegan for single image dehazing," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 825–833.

[126] R. Li, J. Pan, Z. Li, and J. Tang, "Single image dehazing via conditional gen-

erative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8202–8211.

[127] Y. Song, Z. He, H. Qian, and X. Du, "Vision transformers for single image dehazing," *IEEE Transactions on Image Processing*, 2023.

[128] Z. Sun, C. Liu, H. Qu, and G. Xie, "A novel effective vehicle detection method based on swin transformer in hazy scenes," *Mathematics*, vol. 10, no. 13, p. 2199, 2022.

[129] P. Dong and B. Wang, "Transra: transformer and residual attention fusion for single remote sensing image dehazing," *Multidimensional Systems and Signal Processing*, vol. 33, no. 4, pp. 1119–1138, 2022.

[130] S. Chen, T. Ye, J. Shi, Y. Liu, J. Jiang, E. Chen, and P. Chen, "Dehrformer: Real-time transformer for depth estimation and haze removal from varicolored haze scenes," *arXiv preprint arXiv:2303.06905*, 2023.

[131] Y. Zhou, Z. Chen, R. Li, B. Sheng, L. Zhu, and P. Li, "Eha-transformer: Efficient and haze-adaptive transformer for single image dehazing," in *The 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2022, pp. 1–8.

[132] C. Wang, J. Pan, W. Lin, J. Dong, and X.-M. Wu, "Selfpromer: Self-prompt dehazing transformers with depth-consistency," *arXiv preprint arXiv:2303.07033*, 2023.

[133] M. Tong, Y. Wang, P. Cui, X. Yan, and M. Wei, "Semi-uformer: Semi-supervised uncertainty-aware transformer for image dehazing," *arXiv preprint arXiv:2210.16057*, 2022.

[134] Z. Lin, H. Wang, and S. Li, "Pavement anomaly detection based on transformer and self-supervised learning," *Automation in Construction*, vol. 143, p. 104544, 2022.

[135] R. Cong, H. Sheng, D. Yang, Z. Cui, and R. Chen, "Exploiting spatial and angular correlations with deep efficient transformers for light field image super-resolution," *IEEE Transactions on Multimedia*, 2023.

[136] H. Yu, J. Huang, K. Zheng, M. Zhou, and F. Zhao, "High-quality image dehazing with diffusion model," *arXiv preprint arXiv:2308.11949*, 2023.

[137] J. Wang, S. Wu, K. Xu, and Z. Yuan, "Frequency compensated diffusion model for real-scene dehazing," *arXiv preprint arXiv:2308.10510*, 2023.

[138] Z. Luo, F. K. Gustafsson, Z. Zhao, J. Sjölund, and T. B. Schön, "Refusion: Enabling large-size realistic image restoration with latent-space diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1680–1691.

[139] O. Özdenizci and R. Legenstein, "Restoring vision in adverse weather conditions with patch-based denoising diffusion models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[140] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[141] Y. Lin, P. Wang, and M. Ma, "Intelligent transportation system (its): Concept, challenge and opportunity," in *2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*. IEEE, 2017, pp. 167–172.

[142] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.

[143] B. Ji, X. Zhang, S. Mumtaz, C. Han, C. Li, H. Wen, and D. Wang, "Survey on the internet of vehicles: Network architectures and applications," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 34–41, 2020.

[144] Z. Lu, G. Qu, and Z. Liu, "A survey on recent advances in vehicular network security, trust, and privacy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 760–776, 2018.

[145] S. Sharma and B. Kaushik, "A survey on internet of vehicles: Applications, security issues & solutions," *Vehicular Communications*, vol. 20, p. 100182, 2019.

[146] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: A survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2021.

[147] Aptiv. (2023) What is an electronic control unit? Aptiv. [Online]. Available: https://www.aptiv.com/en/insights/article/what-is-an-electronic-control-unit

[148] K. Cheng, Y. Bai, Y. Zhou, Y. Tang, D. Sanan, and Y. Liu, "Caneleon: Protecting can bus with frame id chameleon," *IEEE Transactions on Vehicular technology*, vol. 69, no. 7, pp. 7116–7130, 2020.

[149] E. Seo, H. M. Song, and H. K. Kim, "Gids: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2018, pp. 1–6.

[150] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3893–3902, 2018.

[151] S. Malik, P. Bandi, and W. Sun, "An experimental study of denial of service attack against platoon of smart vehicles," in *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2021, pp. 23–30.

[152] M. Sadaf, Z. Iqbal, Z. Anwar, U. Noor, M. Imran, and T. R. Gadekallu, "A novel framework for detection and prevention of denial of service attacks on autonomous vehicles using fuzzy logic," *Vehicular Communications*, p. 100741, 2024.

[153] D. Swessi and H. Idoudi, "Comparative study of ensemble learning techniques for fuzzy attack detection in in-vehicle networks," in *International Conference on Advanced Information Networking and Applications*. Springer, 2022, pp. 598–610.

[154] F. Martinelli, F. Mercaldo, V. Nardone, and A. Santone, "Car hacking identification through fuzzy logic algorithms," in *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–7.

[155] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, 2020.

[156] Z. Khan, M. Chowdhury, M. Islam, C.-Y. Huang, and M. Rahman, "Long short-term memory neural network-based attack detection model for in-vehicle network security," *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1–4, 2020.

[157] S. Almutlaq, A. Derhab, M. M. Hassan, and K. Kaur, "Two-stage intrusion detection system in intelligent transportation systems using rule extraction

methods from deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[158] I. A. Khan, N. Moustafa, D. Pi, W. Haider, B. Li, and A. Jolfaei, "An enhanced multi-stage deep learning framework for detecting malicious activities from autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 469–25 478, 2021.

[159] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K.-M. Su, and K. A. Robbins, "The prep pipeline: standardized preprocessing for large-scale eeg analysis," *Frontiers in Neuroinformatics*, vol. 9, 2015. [Online]. Available: https://www.frontiersin.org/article/10.3389/fninf.2015.00016

[160] V. R. J. D. 6, "Difference between fir filter and iir filter (with comparison chart)," Feb 2021. [Online]. Available: https://circuitglobe.com/difference-between-fir-filter-and-iir-filter.html

[161] R. Merry, "Wavelet theory and applications: a literature study," *DCT rapporten*, vol. 2005, 2005.

[162] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[163] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach," *PLOS ONE*, vol. 14, no. 5, 2019.

[164] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[165] S. Morshad, M. R. Mazumder, and F. Ahmed, "Analysis of brain wave data using neurosky mindwave mobile ii," in *Proceedings of the International Conference on Computing Advancements*, 2020, pp. 1–4.

[166] A. Karulis *et al.*, "Pybluez: Bluetooth python extension module," 2022. [Online]. Available: https://github.com/pybluez/pybluez

[167] A. Chougule, J. Shah, V. Chamola, and S. Kanhere, "Enabling safe its: Eeg-based microsleep detection in vanets," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[168] L. Wang, Y. Wang, L. Shi, and H. Xu, "Analysis of risky driving behaviors among bus drivers in china: The role of enterprise management, external

environment and attitudes towards traffic safety," *Accident Analysis & Prevention*, vol. 168, p. 106589, 2022.

[169] T. Ito and R. Kaneyasu, "Predicting traffic congestion using driver behavior," *Procedia computer science*, vol. 112, pp. 1288–1297, 2017.

[170] D. Stokols, R. W. Novaco, J. Stokols, and J. Campbell, "Traffic congestion, type a behavior, and stress." *Journal of Applied Psychology*, vol. 63, no. 4, p. 467, 1978.

[171] A. F. Abidin, M. Kolberg, and A. Hussain, "Integrating twitter traffic information with kalman filter models for public transportation vehicle arrival time prediction," *Big-Data Analytics and Cloud Computing: Theory, Algorithms and Applications*, pp. 67–82, 2015.

[172] M. Colesky, K. Demetzou, L. Fritsch, and S. Herold, "Helping software architects familiarize with the general data protection regulation," in *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 2019, pp. 226–229.

[173] B. Gu, A. Xu, Z. Huo, C. Deng, and H. Huang, "Privacy-preserving asynchronous vertical federated learning algorithms for multiparty collaborative learning," *IEEE transactions on neural networks and learning systems*, 2021.

[174] S. Han, W. K. Ng, L. Wan, and V. C. Lee, "Privacy-preserving gradient-descent methods," *IEEE transactions on knowledge and data engineering*, vol. 22, no. 6, pp. 884–899, 2009.

[175] G.-D. Zhang, S.-Y. Zhao, H. Gao, and W.-J. Li, "Feature-distributed svrg for high-dimensional linear classification," *arXiv preprint arXiv:1802.03604*, 2018.

[176] M. Eom and B.-I. Kim, "The traffic signal control problem for intersections: a review," *European transport research review*, vol. 12, pp. 1–20, 2020.

[177] I. Olayode, L. Tartibu, M. Okwu, and d. U. Uchechi, "Intelligent transportation systems, un-signalized road intersections and traffic congestion in johannesburg: A systematic review," *Procedia CIRP*, vol. 91, pp. 844–850, 2020.

[178] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[179] Tzutalin, "Labelimg," Free Software: MIT License, 2015. [Online]. Available: https://github.com/tzutalin/labelImg

[180] Z. Du, L. Qiu, and P. Pisu, "Hierarchical energy management control of connected hybrid electric vehicles on urban roads with efficiencies feedback," in *Dynamic Systems and Control Conference*, vol. 50695. American Society of Mechanical Engineers, 2016, p. V001T16A002.

[181] Z. Du, B. HomChaudhuri, and P. Pisu, "Coordination strategy for vehicles passing multiple signalized intersections: A connected vehicle penetration rate study," in *2017 American control conference (ACC)*. IEEE, 2017, pp. 4952–4957.

[182] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Model predictive control of vehicles on urban roads for improved fuel economy," *IEEE Transactions on control systems technology*, vol. 21, no. 3, pp. 831–841, 2012.

[183] R. O'Toole, "Solving the problem of traffic congestion," *National Center for Policy Analysis. National Center for Policy Analysis*, vol. 8, 2012.

[184] J. Yim, J. Bang, S. Kim, S. Jeong, and Y.-T. Lee, "Uav planning to optimize efficiency of image stitching in disaster monitoring using smart-eye platform," in *2017 International Conference on Platform Technology and Service (PlatCon)*. IEEE, 2017, pp. 1–4.

[185] M. Anbarasan, B. Muthu, C. Sivaparthipan, R. Sundarasekar, S. Kadry, S. Krishnamoorthy, A. A. Dasel *et al.*, "Detection of flood disaster system based on iot, big data and convolutional deep neural network," *Computer Communications*, vol. 150, pp. 150–157, 2020.

[186] M. Qi, S. Cui, X. Chang, Y. Xu, H. Meng, Y. Wang, T. Yin *et al.*, "Multi-region nonuniform brightness correction algorithm based on l-channel gamma transform," *Security and Communication Networks*, vol. 2022, 2022.

[187] D. Julong *et al.*, "Introduction to grey system theory," *The Journal of grey system*, vol. 1, no. 1, pp. 1–24, 1989.

[188] P.-p. Xiong, S. Huang, M. Peng, and X.-h. Wu, "Examination and prediction of fog and haze pollution using a multi-variable grey model based on interval number sequences," *Applied Mathematical Modelling*, vol. 77, pp. 1531–1544, 2020.

[189] Y.-P. Wu, C.-y. Zhu, G.-L. Feng, and B. L. Li, "Mathematical modeling of fog-haze evolution," *Chaos, Solitons & Fractals*, vol. 107, pp. 1–4, 2018.

[190] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[191] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[192] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[193] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[194] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for neural networks for image processing," 2018.

[195] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[196] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016.

[197] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[198] J.-P. Tarel, N. Hautiere, A. Cord, D. Gruyer, and H. Halmaoui, "Improved visibility of road scene images under heterogeneous fog," in *2010 IEEE intelligent vehicles symposium*. IEEE, 2010, pp. 478–485.

[199] J.-P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, "Vision enhancement in homogeneous and heterogeneous fog," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 2, pp. 6–20, 2012.

[200] Y. Han, Z. Liu, S. Sun, D. Li, J. Sun, Z. Hong, and M. H. Ang Jr, "Carla-loc: Synthetic slam dataset with full-stack sensor setup in challenging weather and dynamic environments," *arXiv preprint arXiv:2309.08909*, 2023.

[201] J.-E. Deschaud, "Kitti-carla: a kitti-like dataset generated by carla simulator," *arXiv preprint arXiv:2109.00892*, 2021.

[202] J. Jang, H. Lee, and J.-C. Kim, "Carfree: Hassle-free object detection dataset generation using carla autonomous driving simulator," *Applied Sciences*, vol. 12, no. 1, p. 281, 2021.

[203] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1956–1963.

[204] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4780–4788.

[205] L.-A. Tran, S. Moon, and D.-C. Park, "A novel encoder-decoder network with guided transmission map for single image dehazing," *Procedia Computer Science*, vol. 204, pp. 682–689, 2022.

[206] Y. Jin, W. Yan, W. Yang, and R. T. Tan, "Structure representation network and uncertainty feedback learning for dense non-uniform fog removal," 2022.

[207] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "Dehazenet: An end-to-end system for single image haze removal," *IEEE transactions on image processing*, vol. 25, no. 11, pp. 5187–5198, 2016.

[208] S. Chen, Y. Chen, Y. Qu, J. Huang, and M. Hong, "Multi-scale adaptive dehazing network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[209] A. Chougule, V. Chamola, A. Sam, F. R. Yu, and B. Sikdar, "A comprehensive review on limitations of autonomous driving and its impact on accidents and collisions," *IEEE Open Journal of Vehicular Technology (OJVT)*, 2023.

[210] V. Chamola, A. Chougule, A. Sam, A. Hussain, and F. R. Yu, "Overtaking mechanisms based on augmented intelligence for autonomous driving: Datasets, methods, and challenges," *IEEE Internet of Things Journal*, 2024.

Publications

Journal Papers

1. **Chougule, Amit**, Varun Kohli, Vinay Chamola, and Fei Richard Yu. "Multi-branch reconstruction error (mbre) intrusion detection architecture for intelligent edge-based policing in vehicular ad-hoc networks", *IEEE Transactions on Intelligent Transportation Systems*, 2022.
   DOI: https://doi.org/10.1109/TITS.2022.3201548

2. **Chougule, Amit**, Ishan Kulkarni, Tejasvi Alladi, Vinay Chamola, and Fei Richard Yu. "HybridSecNet: In-Vehicle Security on Controller Area Networks through a Hybrid Two-Step LSTM-CNN Model", *IEEE Transactions on Vehicular Technology*

3. **Chougule, Amit**, Kartik Agrawal, and Vinay Chamola. "SCAN-GAN: Generative Adversarial Network Based Synthetic Data Generation Technique for Controller Area Network." *IEEE Internet of Things Magazine* 6, no. 3 (2023): 126-130.
   DOI: https://doi.org/10.1109/IOTM.001.2300013

4. **Chougule, Amit**, Agneya Bhardwaj, Vinay Chamola, and Pratik Narang. "Agd-net: Attention-guided dense inception u-net for single-image dehazing." Springer, Cognitive Computation 16, no. 2 (2024): 788-801.
   DOI: https://doi.org/10.1007/s12559-023-10244-2

5. **Chougule, Amit**, Jash Shah, Vinay Chamola, and Salil Kanhere. "Enabling Safe ITS: EEG-Based Microsleep Detection in VANETs", *IEEE Transactions on Intelligent Transportation Systems*,2022.
   DOI: https://doi.org/10.1109/TITS.2022.3230259

6. **Chougule, Amit**, Vinay Chamola, Vikas Hassija, Pranav Gupta, and F. Richard Yu. "A Novel Framework for Traffic Congestion Management at Intersections Using Federated Learning and Vertical partitioning.", *IEEE Transactions on Consumer Electronics*, 2023.
   DOI: https://doi.org/10.1109/TCE.2023.3320362

7. **Chougule, Amit**, Vinay Chamola, Aishwarya Sam, Amir Hussain and F. Richard Yu, and Biplab Sikdar. "A Comprehensive Review on Limitations

of Autonomous Driving and its Impact on Accidents and Collisions", *IEEE Open Journal of Vehicular Technology*.
DOI: https://doi.org/10.1109/OJVT.2023.3335180

Biographies

Brief Biography of the Candidate

Amit Chougule is currently pursuing his Ph.D. in Electrical & Electronics Engineering at BITS-Pilani, Pilani Campus, India. He completed his M.Tech at PES University, Bangalore, India, in 2020. He has also worked as a Visiting Researcher at the Carleton School of Information Technology and the Department of Systems and Computer Engineering at Carleton University, Ottawa, Canada. Amit has experience as an artificial intelligence researcher in the healthcare domain, having worked at companies such as Sony Research, Philips Healthcare, and AIvolved Technologies. His research primarily focuses on developing artificial intelligence solutions for autonomous driving and healthcare using computer vision and deep learning.

Brief Biography of the Supervisor

Vinay Chamola received the B.E. degree in electrical and electronics engineering and master's degree in communication engineering from the Birla Institute of Technology and Science, Pilani, India, in 2010 and 2013, respectively. He received his Ph.D. degree in electrical and computer engineering from the National University of Singapore, Singapore, in 2016. In 2015, he was a Visiting Researcher with the Autonomous Networks Research Group (ANRG), University of Southern California, Los Angeles, CA, USA. He also worked as a post-doctoral research fellow at the National University of Singapore, Singapore where he worked in the area of IoT. He is currently an Associate Professor with the Department of Electrical and Electronics Engineering, BITS-Pilani, Pilani Campus where he heads the IoT Research Group / Lab. He has over 45 publications in high ranked SCI Journals including more than 25 *IEEE Transaction* and Journal articles. His works have been published in Journals like *IEEE Transactions on Communications*, *IEEE Transactions on Vehicular Technology*, *IEEE Journal on Selected Areas in Communications*, *IEEE Communications Magazine* etc. Furthermore, his works have been accepted and presented in reputed conferences like *IEEE INFOCOM, IEEE GLOBECOM, IEEE ICC, IEEE PerCom* to name a few. He serves as a reviewer for several IEEE/Elsevier Journals. His research interests include IoT Security, Blockchain, 5G network management and addressing research issues in VANETs and UAV networks. He is

an Associate Editor of the *Ad Hoc Networks* journal, Elsevier and the *IET Quantum Communications* journal. He is a Senior Member of the IEEE.

Brief Biography of the Co-Supervisor

Pratik Narang received the bachelor's and Ph.D. degrees from the Birla Institute of Technology and Science (BITS), Pilani, India, in 2011 and 2016, respectively. In 2014, he was a Visiting Researcher with the New York University (Abu Dhabi campus). He also worked as a Post-Doctoral Research Fellow at the National University of Singapore, Singapore. He is currently an Associate Professor at the Department of CSIS, BITS. His research focuses on building systems using Artificial Intelligence and Machine Learning to improve the existing technologies in Cyber security, Computer Vision, and Healthcare. He was a executed multiple Sponsored R&D projects (funded by Govt. of India and Industrial organizations) in the area of Applied Deep Learning and Artificial Intelligence and also has collaborations with leading universities (NUS, CMU, UIC, etc.) and Industrial organizations (IBM, NVIDIA, Google, etc.) across the world. He is the Faculty Advisor for the Google Developer Student Clubs, BITS, Pilani.