

University of Windsor

## Scholarship at UWindor

---

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

---

10-1-2021

### Emergency Evaluation in Connected and Automated Vehicles

Elvin Eziama  
*University of Windsor*

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>



Part of the [Artificial Intelligence and Robotics Commons](#)

---

#### Recommended Citation

Eziama, Elvin, "Emergency Evaluation in Connected and Automated Vehicles" (2021). *Electronic Theses and Dissertations*. 8776.

<https://scholar.uwindsor.ca/etd/8776>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email ([scholarship@uwindsor.ca](mailto:scholarship@uwindsor.ca)) or by telephone at 519-253-3000ext. 3208.

# Emergency Evaluation in Connected and Automated Vehicles

by

Elvin Eziama

*A Dissertation*

*Submitted to the Faculty of Graduate Studies through  
the Department of Electrical and Computer Engineering  
in Partial Fulfillment of the Requirements for  
the Degree of Doctor of Philosophy at the  
University of Windsor*

*Windsor, Ontario, Canada  
2021*

© 2021 Elvin Eziama

# Emergency Evaluation in Connected and Automated Vehicles

by

Elvin EZIAMA

APPROVED BY:

---

F. Gebali

University of Victoria

---

Z. Kobti

School of Computer Science

---

M. Mirhassani

Department of Electrical and Computer Engineering

---

H. Wu

Department of Electrical and Computer Engineering

---

K. Tepe, Advisor

Department of Electrical and Computer Engineering

September 29, 2021

---

# Declaration of Co-Authorship / Previous Publications

---

## I. Co-authorship

I hereby declare that this thesis incorporates material that is a result of joint research. This thesis contains the outcome of research undertaken by me under the supervision of Dr. K.E. Tepe. The collaboration is covered in Chapter 2,3,4 and 5 of the thesis. The anomaly/attack models and simulation results presented in chapter 5 resulted from joint research undertaken in collaboration with my colleagues, Dr. Danilo Roberto Corral De Witt, Dr. Ahmed Sabbir, Dr. Farooq Awin and my supervisor Dr. Kemal Tepe. In all cases, the key ideas, primary contributions, experimental design of anomaly/attacks, data analysis and interpretations, were performed by the author of this thesis, and the contribution of co-authors were primarily through the provision of valuable suggestions, monitoring and manuscript structure checking.

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

## **II. Previous Publications**

This thesis includes parts from four original papers that have been published and submitted for publication in peer-reviewed conferences and journals. Details of these parts are as follows:

Thesis Chapter	Publication Title/Full Citation	Publication Status
Parts of Chapters 1, 2, and 3	<b>Elvin E.,</b> Saneeha A., Sabbir A., Faroq A. & Kemal, T. (2019, December). Detection of Adversary Nodes in Machine-To-Machine Communication Using Machine Learning Based Trust Model. 2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT).	Published
Parts of Chapters 3 and 4	<b>Elvin E.,</b> Kemal T., Ali B., Kenneth S. N. & Luz M. S. J. (2018, August). Malicious Node Detection in Vehicular Ad hoc Network Using Machine Learning and Deep Learning. 2018 IEEE Globecom Workshops (GC Wkshps).	Published
Parts of Chapter 2	<b>Elvin E.,</b> Luz M.S Jaimes, James, A., Kenneth Sorle N., Ali B. & Kemal T. (2018, December). Machine Learning-Based Recommendation Trust Model for Machine-to-Machine Communication (ISSPIT) (pp. 252–257). IEEE.	Published
Parts of Chapters 4 and 6	<b>Elvin E.,</b> Faroq, A., Sabbir, A., Luz Marina Santos J., Akinyemi P. & Danilo C. (2020). Detection and Identification of Malicious Cyber-Attacks in Connected and Automated Vehicles' Real-Time Sensors. MDPI Applied Sciences.	Published

This is to certify that written permission to include the above-published materials in this thesis has been obtained from copyright owner(s). I certify that the above-mentioned material describes the work completed during my term as a graduate student at the University of Windsor.

### **III. General**

I declare that, to the best of my knowledge, this thesis neither infringes upon anyone's copyright nor violates any proprietary rights and that any ideas, techniques, quotations, or any other material taken from other studies, published or otherwise, have been fully acknowledged in accordance with the standard referencing practices. Furthermore, I have included copyrighted material that surpasses the bounds of fair dealing within the context of the Canada Copyright Act, and I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in this thesis.

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

---

# Abstract

---

An intelligent transportation system (ITS) provides improved transport efficiency and safety based on vehicle communication. Connected and automated vehicles (CAVs) as part of an ITS are projected to revolutionize the transportation industry, primarily by allowing real-time and seamless information exchange between vehicles and roadside infrastructure. Although these CAVs are expected to offer vast benefits, new problems in terms of safety, security, and privacy will also emerge. Since CAVs continue to rely heavily on vehicle sensors and information obtained from other vehicles and roadside units, abnormal sensors and malicious cyber attacks can lead to destructive results and fatal crashes. Therefore, ensuring reliable and secure information dissemination across vehicles and roadside units is vital for many applications and in the safety-critical aspect of CAVs. As a result, mechanisms that can detect anomalies and identify attack sources in real-time are necessary before the mass deployment of CAVs. This dissertation designs an approach for anomaly detection by utilizing deep Learning (DL), and machine learning (ML) mechanisms, namely Bayesian deep learning (BDL) empowered with discrete wavelet transform (DWT), to detect and identify abnormal behavior in CAVs. The proposed approach's numerical experiment shows high performance in detecting anomalies and identifying their scores with high accuracy, sensitivity, precision, and  $F1 - score$ . Furthermore, this proposed method outperforms baseline BDL and convolutional neural network (CNN) approaches in detecting and identifying



anomalies. Performance-wise, the proposed approach is evaluated in terms of the following performance metrics: sensitivity, precision, and  $F_1$  – score. Based on the simulation, the proposed approach achieves performance gains of 6.98 %, 9.10 %, and 7.37 % over CNN and 11.89 %, 7.32 %, and 9.37 % over BDL at duration  $d = 3$  and  $linspace(0, 6000)$  for the difficult gradual drift anomaly.

In another work, a new architecture of ML-Based Trust (MLBT) mechanism in detecting adversary behaviors in a vehicular-based M2M-C (VBM2M-C) framework is proposed. A combination of extreme Gradient Boost (XGBoost) and binary particle swarm optimization (BPSO) is introduced to detect and identify malicious behaviors within the network. The proposed MLBT is evaluated over different probabilities of attacks. The results of this evaluation show that the proposed approach outperforms the state-of-the-art mechanisms by 10 % inaccuracy, 9 % in true positive rate (tpr), and lowers false positive rate (fpr) by 9 %, 10 % in precision, 8.10 % in recall, 9.3 % in sensitivity, and 10 % in  $F_1$  – score with reference to the attacker density of 30 % in the selected metrics better than the compared approaches.

Moreover, an innovative data-driven approach was equally developed, which involves the combination of discrete wavelet transform (DWT) and double deep Q network (DDQN) method for anomaly detection in CAVs. The DDQN is modified to accommodate classification by taking the state’s data feature while labeling as the action. The features in DWT and DDQN are combined to enhance anomaly detection performance in CAV networks. The DWT smoothens the basic safety messages (BSMs) sensor reading before the BSMs are fed into the DDQN approach.  $F_1$  – score and sensitivity are used to access the performance of the proposed method. Overall, the proposed method achieves a performance gain of 20 % and 10 % at a small density of anomaly distribution and 12 % and 8 % at a high density of anomaly distribution for ensemble multilayer perceptron (EMLP) and support vector machine (SVM).

---

# Dedication

---

Every challenging work needs self-efforts as well as the guidance of people close to our hearts. I dedicate this thesis to my wife, Sandra Eziama; kids, Elvin Jnr Chinemezu C. and Eric Ugonna C. Eziama, siblings, Henry Osineke, Queenelia Chioma, Henriatta Ekeoma, Stella Chisom, and Judith Ngozi Eziama; parents, Hon. Chief Vincent N. and Joyce C. Eziama; and family and friends whose affection, love, encouragement, and prayers enabled me to get such success and honor. Thank you everyone for being with me.

---

# Acknowledgements

---

First of all, I would like to express my deepest appreciation to my supervisor, Dr Kemal E. Tepe, for his patience, motivation, encouragement, and invaluable advice and support, without which I could not have completed this course. Moreover, I would like to thank the rest of my thesis committee, Dr Ziad Kobti, Dr Mitra Mirhassani, and Dr Huapeng Wu, for serving on my committee and helping me with their valuable suggestions and contributions regarding the improvement of this work. Further, I would like to acknowledge the assistance I received from many teachers, lecturers, authorities, technical team, and all the administrative personnel of the University of Windsor, especially from Ms. Andria Ballo, for her support and advice during the development of this research, and Dr. Ali Balador, Dr. Luz Marina Santos-Jaimes, Dr Kenneth Sorle Nwizege, and Dr. James Agajo, for their support in seeing this study and research Ph.D. study a mission accomplished. Last but not least, I would like to thank all the fellow graduate students and colleagues of the WiCIP Research Lab at the University of Windsor, especially Dr. Danilo Corral-De-Witt, Dr. Sabbir Ahmed, Dr. Saneeha Ahmed, Dr. Faroq Awin, and all the members of the lab, for sharing their thoughts, ideas, and having a wonderful time throughout this research.

---

# Contents

---

<b>Declaration of Co-Authorship / Previous Publications</b>	<b>iii</b>
<b>Abstract</b>	<b>vii</b>
<b>Dedication</b>	<b>ix</b>
<b>Acknowledgements</b>	<b>x</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xviii</b>
<b>List of Abbreviations</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Trust Related Challenges in CAV . . . . .	2
1.2 Motivation . . . . .	4
1.3 Problem Statement . . . . .	4
1.4 Research Objectives . . . . .	6
1.5 Research Contributions . . . . .	7
1.6 Organization of the Thesis . . . . .	8
<b>2 Literature Review on Detection Approaches and Related Works</b>	<b>9</b>
2.1 Connected Autonomous Vehicles . . . . .	9
2.1.1 Architecture of CAVs . . . . .	10

Challenges . . . . .	12
2.1.2 Dedicated Short-Range Communication . . . . .	16
2.1.3 Cellular Vehicle to Everything . . . . .	16
2.1.4 ML Detection Approach . . . . .	17
2.1.5 Different Types of Machine Learning . . . . .	18
2.2 Related Work . . . . .	19
<b>3 Evaluation Methodology for Misbehavior Formulation and Detection</b>	<b>23</b>
3.1 Methodology . . . . .	23
3.1.1 Studied Attacker Models . . . . .	24
3.1.2 CAVs Data Characteristics and Descriptions . . . . .	24
3.1.3 Attack/Anomaly Model for CAV System . . . . .	26
3.1.4 Misbehavior Scenarios and Alert Types . . . . .	28
Emergency Electronic Brake Light (EEBL) . . . . .	28
Change of Lane (CoL) . . . . .	29
Path Deviation Alert (PDA) . . . . .	29
3.1.5 Attacker Model for Machine-to-Machine (M2M) Commu- nication . . . . .	30
3.1.6 Evaluation Strategy . . . . .	31
3.1.7 Misbehavior Detection Metrics . . . . .	32
<b>4 Misbehavior Detection in CAV Network Based on Machine Learning</b>	<b>35</b>
4.1 M2M Attack Detection Approach . . . . .	35
Binary Particle Swarm Optimization (BPSO) . . . . .	35
Extreme Gradient Boosting (XGBoost) . . . . .	37
Random Forest . . . . .	40
4.1.1 The Proposed Approach . . . . .	41
4.1.2 CAV Attack Detection Mechanism . . . . .	44
Convolutional Neural Network (CNN) . . . . .	44

	Classification Criterion of Convolutional Neural Network (CNN) Algorithm . . . . .	47
4.1.3	Discrete Wavelet Transform (DWT) . . . . .	47
	Bayesian Deep Learning (BDL) . . . . .	49
4.1.4	Discrete Wavelet-Based Deep Reinforcement Learning with Double Q Learning (DWT-DDQN) . . . . .	54
4.1.5	Support Vector Machine (SVM) . . . . .	58
4.1.6	Ensemble Multi layer Perception (EMLP) . . . . .	58
4.1.7	Bayesian Deep Learning-Empowered Discrete Wavelet Transform (DWT-BDL) . . . . .	60
	Classification Criterion of the Proposed Approach . . . . .	61
<b>5</b>	<b>Results and Discussion</b>	<b>64</b>
5.1	Results: Machine-to-Machine (M2M) Model Detection Perfor- mance . . . . .	64
	Discrete Wavelet Transform (DWT) Pre-Analysis of the Data	67
5.2	Results and Discussion . . . . .	69
5.2.1	Comparison of the Proposed DWT-BDL Approach and Conventional Approaches Under Single Anomaly System	69
	Impact of Network Density on Anomaly Detection . . . . .	70
	Impact of Attack Duration . . . . .	77
	Bias Anomaly Setting . . . . .	77
	Gradual Drift Anomaly Setting . . . . .	81
5.2.2	Performance of the Approaches During the Multiple Anomaly Scenario . . . . .	87
5.2.3	Approaches Under Single Anomaly System Under DWT- BDL Proposed Approach . . . . .	94
5.2.4	Approaches Under Multiple Anomaly System Under DWT-DDQN Proposed Approach . . . . .	98

<b>6 Conclusion and Future Work</b>	<b>104</b>
6.1 Conclusion . . . . .	104
6.2 Future Work . . . . .	108
<b>References</b>	<b>109</b>
<b>Appendix</b>	<b>119</b>
<b>A Copyright Permissions</b>	<b>120</b>
<b>B Mathematical Notations</b>	<b>124</b>
<b>C Mathematical Derivations and Illustrations</b>	<b>125</b>
<b>D Key Literature Related to Anomaly Detection Approaches in CAV</b>	<b>127</b>
<b>Vita Auctoris</b>	<b>128</b>

---

# List of Figures

---

2.1	The basic system architecture of connected vehicles having three types of communications: vehicle to vehicle (V2V), infrastructure to infrastructure (I2I), and infrastructure to vehicle (I2V; figure adapted from [31]) . . . . .	12
2.2	Network and attacker behavior in the time frame $T_1$ . . . . .	14
2.3	Network and attacker behavior in the time frame $T_2$ . . . . .	15
2.4	States of an attacker . . . . .	16
3.1	False emergency electric brake light (EEBL) alert . . . . .	28
3.2	Change of lane (CoL) attack scenario . . . . .	29
3.3	Path deviation attacker (PDA) Scenario . . . . .	30
4.1	One-dimensional convolutional neural network (CNN-1D) architecture . . . . .	46
4.2	Wavelet threshold denoising mechanism . . . . .	49
4.3	Bayesian hierarchical framework for neural network . . . . .	53
4.4	Basic safety message (BSM) dataset preparation for the training of the double deep Q-network (DDQN) approaches Source: Adapted from [78] . . . . .	55
4.5	Double deep Q network (DDQN) approach training scheme Source: Adapted from [78] . . . . .	57
4.6	Bayesian hierarchical framework for neural network . . . . .	63



5.1	Accuracy vs. attacker percentage scenario . . . . .	66
5.2	True positive rate vs. attacker densities . . . . .	66
5.3	False positive rate vz. attacker densities . . . . .	67
5.4	Detection performance of CNN, BDL, and the proposed approach during the instance anomaly scenario . . . . .	72
5.5	Detection performance of CNN, BDL, and the proposed approach during the bias anomaly scenario . . . . .	74
5.6	Detection performance of the CNN, BDL, and the proposed approach during the gradual drift anomaly scenario . . . . .	76
5.7	Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions during the bias anomaly scenario . . . . .	78
5.8	Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions during the bias anomaly scenario . . . . .	79
5.9	Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions in bias anomaly scenario . . . . .	80
5.10	Detection performance at 95% CI and CRI across 15 to 20 different executions for all three approaches, at anomaly rate $\eta = 50\%$ and in the presence of all the types of anomalies . . . . .	89
5.11	Detection performance at 95% CI and CRI across 15 to 20 different executions for all three approaches, at anomaly rate $\eta = 10\%$ and in the presence of all the types of anomalies . . . . .	91
5.12	Performance variation of the various approaches trained on bias anomaly during the gradual drift anomaly, at $\eta = 10\%$ and $\eta = 50\%$ . . . . .	93
5.13	Comparison of rewards and losses at different episodes of DDQN training at various $\gamma$ values . . . . .	101

5.14	Performance variation of the approaches during the gradual drift anomaly scenario, at $\eta = 10\%$ and $\eta = 50\%$ . . . . .	102
5.15	Performance variation of the approaches in terms of <i>F1 – score</i> and sensitivity metrics during the gradual drift anomaly scenario, at $\eta = 10\%$ and $\eta = 50\%$ . . . . .	102

---

# List of Tables

---

4.1	XGBoost optimized parameters . . . . .	42
5.1	Descriptive statistics of selected basic safety message (BSM) variables . . . . .	68
5.2	Detection performance of the approaches during the drift anomaly scenario, with $\mathcal{U}(0, 6000)$ . . . . .	83
5.3	Detection performance of the approaches during the drift anomaly scenario, with $\mathcal{U}(0, 4000)$ . . . . .	84
5.4	Detection performance of the approaches during the drift anomaly scenario, with $\mathcal{U}(0, 2000)$ . . . . .	85
5.5	Detection Performance and the 95 % confidence interval across 10 different executions for the three approaches, at anomaly rate of 10 %, duration 7 with respect to Gradual drift anomaly type with anomaly distribution of $linspace(0, 10000)$ . . . . .	96
5.6	Detection Performance and the 95 % confidence interval across 10 different executions for the three approaches, at anomaly rate of 10 %, duration of 7 with respect to Gradual drift anomaly type with anomaly distribution of $linspace(0, 2000)$ . . . . .	97
5.7	Approach complexity based on training time . . . . .	98

---

# List of Abbreviations

---

**AV** Automated Vehicles

**BSM** Basic Safety Message

**CAN** Controller Area Network

**CAVs** Connected and Automated Vehicles

**CNN** Convolutional Neural Networks

**CNN-ALSTM** Convolutional Neural Network with Attention based Long Short Term Memory

**CTs** Clear to Send

**DBNs** Deep Belief Networks

**DT** Decision Tree

**CoL** Change of Lane

**DCT** Data Centric Trust

**DRSC** Dedicated Short Range Communication

**DWT** Discrete Wavelet Transform

**ECT** Entity Centric Trust

**EEBL** Emergency Electronic Brake Light

**FCC** Federal Communication Commission

**FFNN** Feed Forward Neural Network

**GPS** Global Position Systems

**IDS** Intrusion Detection Systems

**KF-CNN** Kalman Filter based Convolutional Neural Network

**LRR** Long Range Radar

**MAC** Medium Access Control

**NN** Neural Network

**OBU** On Board Unit

**OFDM** Orthogonal Frequency Division Multiplexing

**PHY** Physical Layer

**RFID** Road Frequency Identification

**RSU** Road Side Unit

**SAE** Society of Engineer

**TDMA** Time Division Multiplexing

**USDOT** United Department of Transportation

**V2I** Vehicle to Infrastructure

**V2V** Vehicle to Vehicle

**V2P** Vehicle to Pedestrian

**VANETs** Vehicle Ad hoc Networks

# Introduction

---

Connected and autonomous vehicles (CAVs) as part of intelligent transportation systems (ITSs) are emerging technology, where a large number of vehicles can collect, process, and communicate information through vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, to make collaborative decisions without direct human intervention [1].

CAV is projected to provide significant economic and life-saving benefits to the transportation industry. In the United States (US) alone, as of 2018, traffic accidents caused 36,473 deaths, and traffic congestion contributed to the loss of about 115 billion USD [2]. These traffic accidents and economic loss can be significantly reduced with vehicular communications [3]. Such a high rate of traffic accidents has prompted researchers to make significant efforts to investigate many aspects of vehicular communication [3]. That is why Federal Communication Commission (FCC) has allocated 75 MHz bandwidth for CAV applications in the US. One of the technologies that deliver CAV is dedicated short-range communications (DSRCs), and the other one is cellular-based cellular vehicle to everything (C-V2X). A similar spectrum is allocated to other parts of the world for CAVs. Further, standards such as IEEE 1609.1– IEEE 1609.4 were designed to provide active safety and enhance the driver experience [4].

Those IEEE standards define physical (PHY), medium access control (MAC), and security and application layers [3]. In addition to IEEE standards, the Society of Automotive Engineers (SAE) standards, such as J2735, J2945, and J3161, define the protocol and application layers specifications for CAVs. Among those protocols and messages, one of the most critical ones is basic safety messages (BSMs), which communicate important information about the state of the vehicle, such as acceleration, heading, Global Position System (GPS) coordinates, speed, and braking [5]. BSMs are created at the rate of 10 messages per second (m/s) to accurately alert and warn remote vehicles (RVs) about the state of the host vehicles (HVs) for safety and emergency applications such as forward collision warning.

It has been reported that, any disruption and attacks on these safety messaging applications can potentially have deadly consequences [6]. To prevent CAV applications from those disruptions and attacks, disseminated information should be trustworthy (i.e., correct) and anomaly-free. To authenticate the senders and provide integrity, BSMs must be signed. However, a certified and authorized device might be compromised or tampered by malicious users to deliver inaccurate and malicious information. Even worse, malicious users can group to coordinate their efforts to attack the network to inflict accidents. As of today, identifying a malicious user in these networks is an active research area.

## **1.1 Trust Related Challenges in CAV**

The concept of trust has received extensive attention in different disciplines, such as philosophy, sociology, and politics, as one of the primary factors in decision-making [7], [8]. Trust is defined as the receiving vehicle's opinion on how honest the sending vehicle is in reporting its state. It is considered as soft security since it is subjective and can be described in different ways. For example, a binary trust can be defined as "1" or "0"; a multilevel trust can be

represented as Level 1, Level 2,..., Level n, as values between  $[-1, 1]$ , or as probabilistic measure with values between  $[0, 1]$ .

Trust evaluation is categorized into three models: entity-centric trust (ECT), data-centric trust (DCT), and hybrid model. In the ECT model, trust is defined as an integration of multiple factors about the entity. In other words, trust is established on the vehicles. However, the model has numerous drawbacks simply because of its time-invariant nature [9]. In this case, a valuable amount of time is taken for a receiving vehicle to establish a decision about a given vehicle. Furthermore, ECT involves rounds of complex iterations, resulting in the system's high detection latency. The ECT can be exploited within the dynamic vehicular networks [10], [11].

In the ECT model, a decision is made based on the interaction among vehicles. Decision-making in this context involves the combination of direct trust, recommended trust, and previous experience of the receiving vehicle about a reporting vehicle. Here, decision-making might incur delays since a receiving vehicle only takes a final decision after gathering information from these three sources: direct trust, recommended trust, and previous history.

However, most emerging mobile networking technologies are mainly data-centric [11] since it is relatively easy to establish trust in the data rather than the reporting vehicles. For instance, in a CAV, the vehicle's identity as a security measure in ECT has no contribution in the update of the events and status of the vehicles in the network [10], [11]. DCT uses alert messages like safety warnings, traffic information update, time freshness, and location relevance to provide valuable information about the vehicular network state.

Integrating basic safety message's (BSM's) correlations in DCT will give more insight into malicious activities in the vehicular networks. These BSM correlations are effectively used for trust modeling and the detection of attacks. The numerous attributes to measure trust are speed correlation with break status, vehicular density with speed, distance of observing a vehicle to an event



(such as an accident), and information report of incidence in the network.

## 1.2 Motivation

The CAV communication network is susceptible to severe attacks and privacy challenges. Moreover, robust methods that can effectively identify malicious and inaccurate information in the CAV network have not been well explored. However, since security and reliability are extremely important for the efficiency of CAV applications, CAV safety and emergency applications must receive correct and secure information to operate reliably to achieve their objectives.

Misinformation and disruption in a CAV network may result in accidents, traffic congestion, and many other undesirable consequences. Hence, the dissemination of information must be secure and anomaly-free, so the receiver must trust this to act on it. BSMs are signed to authenticate the sender and to verify the message integrity. However, a certified device might be compromised and tampered with by malicious users. Compromised certified users can send incorrect information to create problems in the communication network. Even worse, malicious users can collaborate to coordinate their efforts to attack the transportation network, thereby inflicting substantial damage.

## 1.3 Problem Statement

CAV network consists of many moving vehicles and is extremely dynamic; therefore, it poses numerous problems for communicating vehicles [12]. Such dynamic network topology results in a partitioned network with isolated vehicles and short-lived links among vehicles and roadside units (RSUs). This

dynamic network coupled with possible malicious vehicles makes it more challenging to estimate correct information and detect attackers at the receiving end [12].

Moreover, vehicles in the CAV network may share information to warn others about their location, work zones, or traffic conditions. This sharing informs neighboring vehicles about the state of the traffic condition. The efficiency of these CAV applications depends on the availability of continuous and reliable information about vehicles' status in terms of parameters such as location, speed, and direction [13].

CAVs use this information for vital decision-making, such as optimizing route, preventing congestion, ensuring safety, and avoiding accidents. However, misbehaving CAVs will often constitute a critical problem in the integrity of such vital decision-making applications. Consequently, relying on those incorrect messages can cause life-threatening cases.

Traditional safety mechanisms, such as cryptographic algorithms, coupled with preventive security countermeasures are inadequate to protect CAVs against attackers [14], as they cannot always guarantee that data are correct [15], [16]. Thus, the identification of misbehavior is one of the most important defense mechanisms that can thwart such threats. Several approaches exist to identify malicious information in CAVs, such as entity-centric (EC) and data-centric (DC) approaches. EC approach is obtained from the past behaviors of the sender vehicle and its reputation among its neighbors [4]. However, this approach has the disadvantages of being time-invariant and slow in decision-making [17].

On the other hand, the DC approach depends on the message's consistency and plausibility [18], [19]. However, most of those proposals are based on static rules built on the predefined context; attackers can easily circumvent those rules and carry out undetected attacks. Moreover, current defense mechanisms are designed for a specific attack and cannot be generalized [20]. Consequently, the

existing solutions for detecting misbehavior suffer from high false alarms and low detection levels due to ignorance of the complexity of the vehicular environment and assumptions that do not reflect the reality of CAV attributes [9].

## 1.4 Research Objectives

As discussed in the Motivation and Problem Statement sections, providing robust approaches to mitigate misbehavior and identify attacks is vital to ensuring the sound and long-term operation of CAVs. Therefore, this research strives to achieve the following goals:

1. Developing and designing efficient misbehavior detection mechanism using machine learning (ML) and discrete wavelet transform (DWT) of time series of data to improve the overall attack/anomaly detection performance of CAV networks.
2. Performing extensive simulation analysis to investigate the effects of anomaly types, density, and duration in single- and multiple-attack/anomaly scenarios in the CAV networks.
3. Presenting a comparative study of different anomaly detection approaches that critically monitor the network in the context of CAV and machine-to-machine M2M technology. Specifically, providing an in-depth evaluation of ML, deep learning (DL), and deep reinforcement learning (DRL) approach to detect misbehavior in CAV and M2M networks.

## 1.5 Research Contributions

This thesis undertakes an end-to-end study on critical safety concerns in CAVs and M2M communication applications after a comprehensive literature review. Anomaly and misbehavior detection ability in a CAV is investigated by providing robust ML and DL approaches. The thesis contributions can be summarized as follows:

1. To the best of our knowledge, we are the first or one of the few that have developed and extensively evaluated single and multiple anomalies/attacks in the CAV context. The multiple anomaly framework is illustrated using a real-life environment where several anomalies/attacks are prevalent.
2. Our proposed approaches are capable of providing robust detection and identification of anomaly and type, and filters noise in the messages.
3. We present a comparative study of different anomaly detection approaches that critically monitor the network in the context of both CAV and M2M. Specifically, we provide an in-depth evaluation of ML, DL, and DRL approaches to detect misbehavior in CAV and M2M networks.
4. We propose an optimized Bayesian deep learning (BDL), double deep Q learning (DDQN), with discrete wavelet transform (DWT) and extreme gradient boost (XGBoost) enhanced binary particle swarm optimization (BPSO), approach that not only provides reliable anomaly detection but also is scalable to the changing density of the CAV and M2M networks.
5. We present extensive simulation results to investigate the effects of varying anomaly distributions, duration, incident rate, and type in a CAV setting.

## **1.6 Organization of the Thesis**

Chapter 2 provides background and review of relevant publications and ideas. Chapter 3 presents the evaluation methodology for misbehavior formulation and detection. Chapter 4 describes the misbehavior detection for CAV and M2M based on ML. Chapter 5 presents the results and discussions. Chapter 6 concludes the thesis and speculates on the future research opportunities.

# Literature Review on Detection Approaches and Related Works

---

CAV security and privacy is a broad topic involving multiple subjects such as the system architecture, deployments and platforms with different requirements, attackers with varying motivations, levels of abilities, and complexities of threats and countermeasures [3].

A comprehensive analysis of CAV security and privacy is required to recognize the most relevant issues and identify avenues for innovation to be embraced and implemented in the automotive industry. Therefore, in this study, relevant issues related to detection mechanisms of traditional security techniques, such as cryptography and DC and EC detection, are reviewed.

## 2.1 Connected Autonomous Vehicles

CAV is one of the key developments in the field of ITS. The CAV communication can provide and enhance safety, environmental sustainability, and user experience. The CAV framework combines the advantages of connected vehicle (CV) and autonomous vehicle (AV) technologies [21].

CAV technology allows the communication and cooperation between vehicles and infrastructure to share vital messages such as speed, location, acceleration, brake condition, and traffic signaling. One of the important messages in V2V is BSMs, which broadcast vehicles' vital information to surrounding vehicles and infrastructure. The V2V communication range is approximately 400 meters at line of sight (LOS) [21]–[25]. V2V will complement currently available AV sensors such as radar and vision to detect other vehicles and identify hazardous conditions.

The movement towards CAVs will eliminate vehicle accidents, reduce injuries and fatalities, improve fuel economy, and provide better traffic flow efficiency. However, the connected computing infrastructure of CAV is likely to be vulnerable to attacks. Recent studies have identified vulnerabilities associated with various sensors, controls, and communication mechanisms [21].

A thorough road test study shows many open problems associated with CAV technology [26], [27]. It is essential that vehicles cooperate among themselves and share reliable information about the current update of events in a CAV network. Further, decisive effort should be made to identify the vulnerabilities in CAV network, determine the risk associated with these networks under cyberattacks and provide mitigation approaches so that future applications will perform resiliently when they are attacked.

### **2.1.1 Architecture of CAVs**

In CAV environments, a vehicle will share its sensor and vehicle dynamics information with other vehicles in its vicinity through BSMs. This information sharing will improve traffic flow, road safety, and fuel economy [28]. A simple illustration of CAV networks is shown in Figure 2.1 and explained as follows:

- (i) V2V Communication: This applies to any vehicle connected to any other type of vehicle. Communication mode is inherently wireless, and messages contain information regarding parameters such as position, travel direction, and speed of the moving vehicle to provide localization. V2V communication technology enables vehicles to transmit and receive omnidirectional messages, usually every 100 milliseconds (ms), and establishes a 360-degree awareness of other neighboring vehicles.

Vehicles may assess possible crash threats through these messages as they evolve. V2V technology uses dedicated short-range communication (DSRC) or C-V2X technologies operated in the 5.9 GHz band [29]. Moreover, vehicles also employ a human-machine interface (HMI) to interact with the driver to warn other road users regarding occurrences such as change speed or change direction, thereby preventing collisions.

- (ii) V2I Communication: This type deals with communication between vehicles and highway infrastructure, including traffic lights, road sensors, speed cameras, traffic sensors, satellite communications, and parking meters. V2I communication is bi-directional and wireless.

V2I communication is carried out using DSRC/C-V2X frequencies equivalent to V2V connectivity. Data collected from infrastructure are used to provide real-time advisory information to the traveler, e.g., adjustment of direction, brake, and routes or escape from other circumstances.

- (iii) V2P Communication: This type deals with the communication of nearby pedestrians' safety to vehicles. The goal is to increase pedestrian safety, prevent collisions affecting road users, and improve vehicle occupants' safety. Pedestrian detection system is an essential feature of V2P [29], [30], which can be applied in a variety of ways: (1) implanted within vehicles



(for blind-spot alert, accident ahead alert, etc.); (2) embedded within roadside facilities, e.g., lane-closing warning; and (3) carried by pedestrians, e.g., smart sensors, to notify drivers.

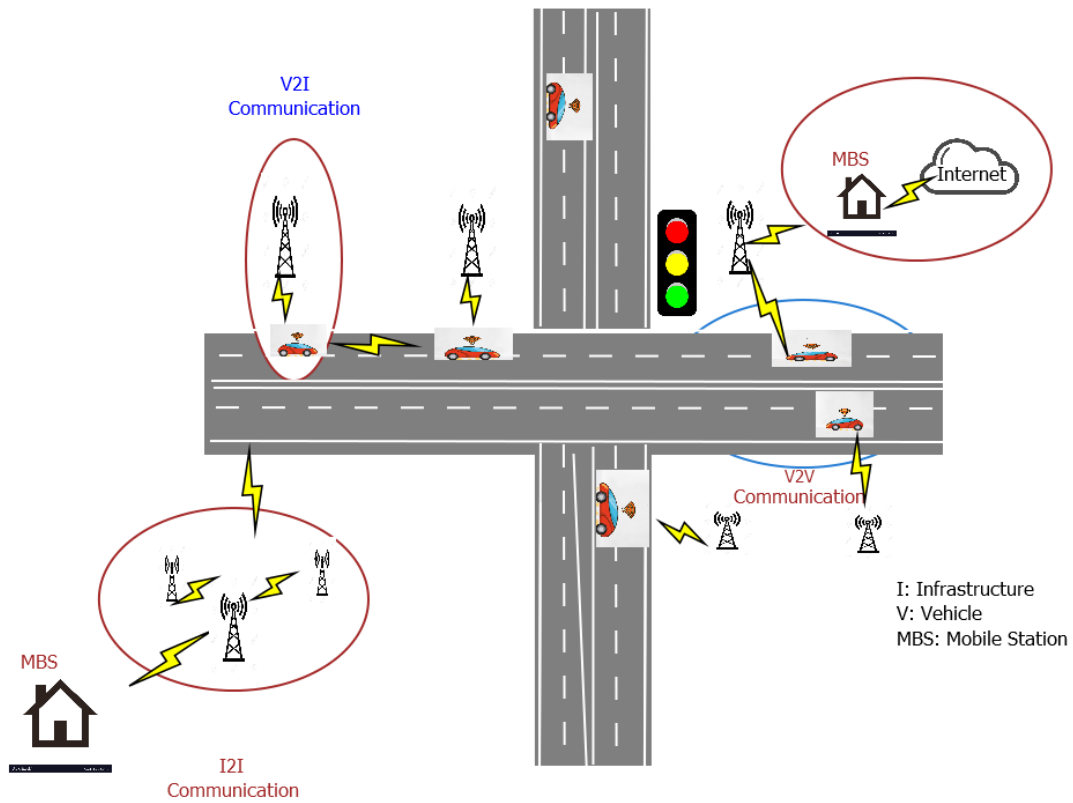


FIGURE 2.1: The basic system architecture of connected vehicles having three types of communications: vehicle to vehicle (V2V), infrastructure to infrastructure (I2I), and infrastructure to vehicle (I2V; figure adapted from [31])

## Challenges

Owing to the movement of vehicles, CAV tends to be highly dynamic [32]. As a result of this dynamic topology, communication among CAVs lasts for a short period, resulting in limited data exchange; in other words, the connectivity is too transient.

Additionally, the various trajectories of travel of each CAV and attackers' behaviors can contribute to the dynamic topology of the CAV network. This complex nature of the topology makes CAVs more vulnerable to attacks [33].

In general, network and attacker behaviors play a major role in CAV network security. These two behaviors are inter-related because the CAV system network demonstrates network entities with positive behavior, while the attacker shows negative behavior. The negative behavior of the attacker poses a threat to the network users. Hence, an attacker's activity is capable of modifying the structure or topology of the network. This modified network topology can make it impossible for the detector approach to capture the behaviors of adversarial vehicles in the CAV setting.

#### **- Network Behavior**

In this context, users, vehicles, and RSUs are the CAV network entities, and each entity has its behavior. Malicious vehicles can modify the topology, which may misguide other entities in the CAV network. The following are the primary reasons for the change in network behavior:

- (i) Speed of vehicles: The speed and movement of vehicles are different relative to mobile ad hoc networks (MANETs), which, if very strong, the vehicle's location in a network will be modified in a few seconds. Such modification makes it difficult for models to capture the dynamics of attacker behaviors.
- (ii) Density of the network: This is another significant aspect of the vehicle environment and deals with the presence of many vehicles on the road. Each vehicle is relatively dynamic with continuously shifting locations. This shifting of location in the CAVs network provides a daunting challenge for effective coordination in a high-density network due to no central control. This situation equally creates challenges regarding identification of attacker vehicles.
- (iii) Dynamics of network topology: The dynamic network topology and the narrow range of V2V communication can result in regular network partitioning and disconnection. Figures 2.2 and 2.3 illustrate the unstable state

of the CAVs network. In this context, new vehicles join the network, which quickly changes the overall network topology.

Figure 2.2 depicts the condition whereby vehicles exchange information on the highway among themselves, and an attacker, vehicle  $X$  launches a malicious attack on Vehicles  $A$  and  $B$  in the time frame  $T_1$ . Further, Figure 2.3 shows the change in the topology of the network in the time frame  $T_2$ .

The vehicle location shift results in attacker Vehicle  $X$ , Vehicles  $A$  and  $B$  being near to RSU. The variation of vehicle location results in the launch of an attack in different positions. As depicted in Figure 2.3, the attack is shifted from vehicles  $A$  and  $B$  to RSU due to variation in attacker Vehicle  $X$ 's old and current positions.

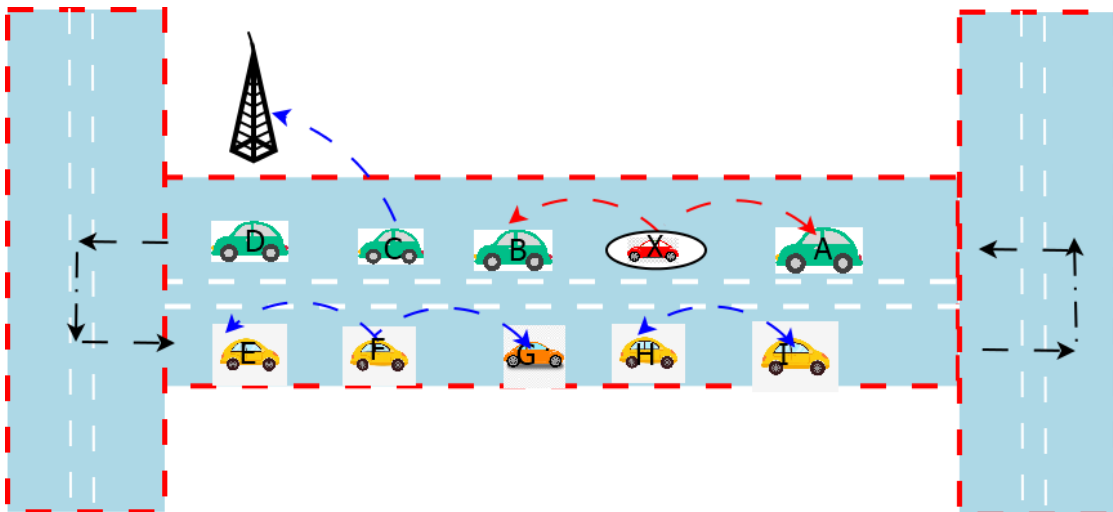


FIGURE 2.2: Network and attacker behavior in the time frame  $T_1$

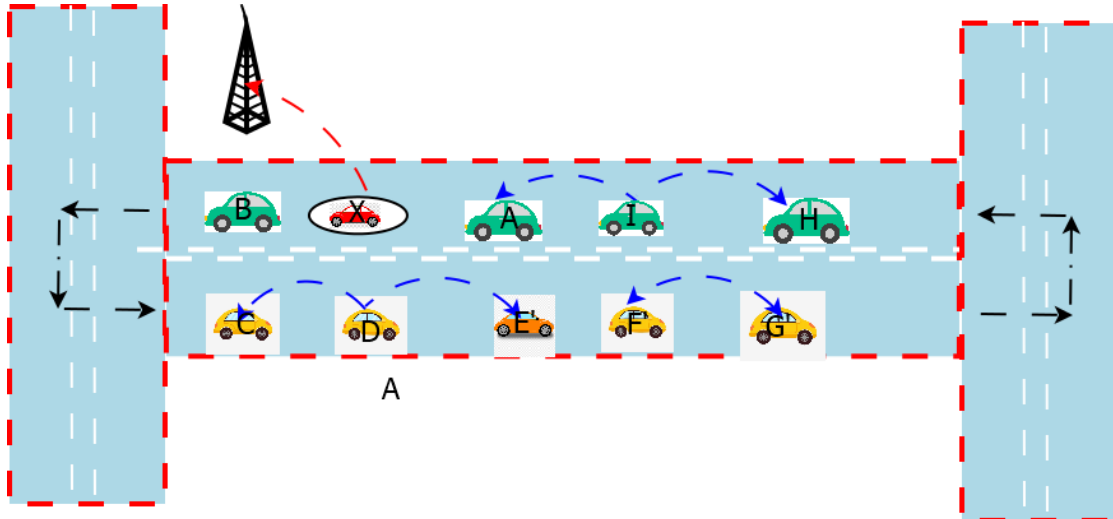


FIGURE 2.3: Network and attacker behavior in the time frame  $T_2$

### - Attacker Behavior

In the simulated network, a vehicle can be in two states, namely honest and attacker. The honest vehicles always provide true information. However, an attacker can switch between honest and attacker states. How often an attacker switches, its state is varied to capture the performance of the designed detection algorithms. This state switching of an attacker makes it difficult to identify it in the network. The two states are depicted in Figure 2.4 and are explained as follows:

**Zero (honest) state:** In this case, the attacker does not initiate an attack in the network, but rather exhibits honest behavior and communicates its correct information.

**One (attack) state:** In this case, the attacker enters into the attacker state and changes its behavior for a certain amount of time and creates an attack.

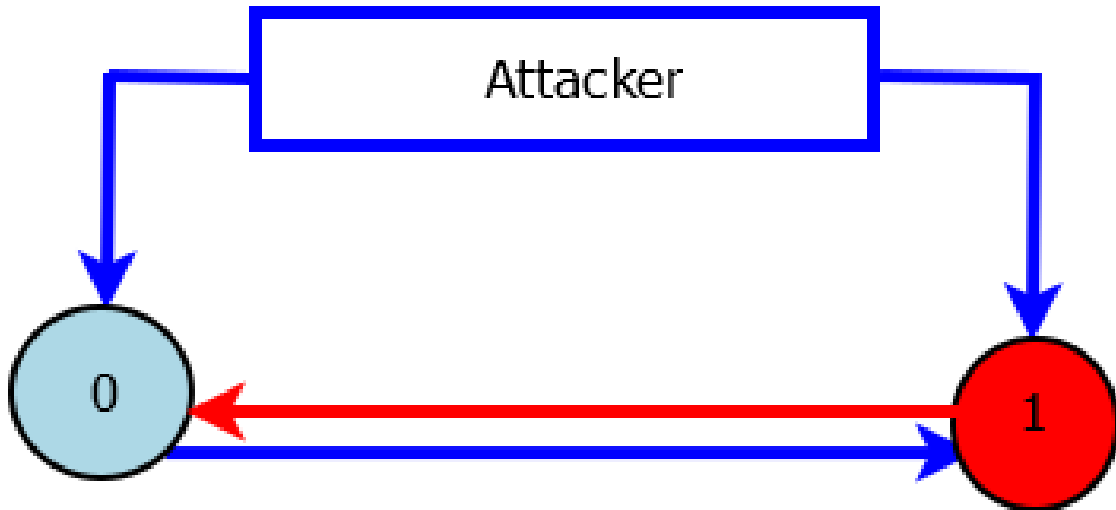


FIGURE 2.4: States of an attacker

### 2.1.2 Dedicated Short-Range Communication

DSRC is the primary enabler of V2V and V2I wireless communication technologies. Initially, the US FCC allocated a 75 MHz bandwidth at 5.9 GHz spectrum band for DSRC operation. DSRC's requirements are published as several IEEE standards that deal with wireless access in vehicular environment (WAVE), which define the PHY and MAC layers (IEEE 802.11p) and the upper layers (standards of IEEE 1609 family). However, in 2020, FCC limited DSRC's bandwidth to 10 MHz and mandated cease of all DSRC operations by 2023.

Although DSRC will not be the V2V technology in the US, several automobile producers, information and communication technology (ICT) providers, academics, and governments are still working closely together to realize this exciting technology in other countries [34], [35], [36].

### 2.1.3 Cellular Vehicle to Everything

C-V2X is a vehicle communication technology focused on a cellular network that meets the requirement of both the initial case of LTE V2X in 3rd Generation Partnership Project (3GPP) Release 14 and the advance case of 5G-based V2X [37]. The 3GPP regulates the C-V2X technology based on the growing needs of

V2X providers and the automotive industry [38]. The communication technology facilitates conventional vehicle-to-network (V2N) communication to provide back-end services and enables direct transmission between end devices, i.e., V2V, V2P, and V2I communication [39]. For standardization, the C-V2X technology was launched as LTE-V in 3GPP Release 14 and is being progressively evolved in 3GPP. It is now part of 3GPP Release 15 to satisfy the criteria for vehicular implementations of fifth-generation (5G) cellular networks. The side-link communication of C-V2X allows devices to work practically in two modes (such as in in-coverage and out-of-coverage modes).

C-V2X addresses use cases for traffic quality and safety, where data are communicated with short latency constraints in the vehicle's vicinity. The safety and traffic efficiency applications of C-V2X depend on a periodic exchange of low latency messages [39].

#### **2.1.4 ML Detection Approach**

This section highlights some of the main problems associated with CAVs. The section also offers a detailed description of how ML resolves these challenges, in particular DL algorithms.

ML algorithms emphasize the ability to deal with changes and challenges and respond to the environment through adaptive data-driven decision-making [40], [41]. ML attribute is quite unlike traditional schemes, which rely on explicit device parameters and thresholds for decision-making. The ML algorithms can take advantage of several features of generated data, such as BSMs (i.e., vehicle activity patterns, vehicle locations/kinetics, etc.), to learn the system dynamics and then select correct features to establish a learning process. The information from BSMs can further be used to improve network performance and detection accuracy of the effects/impacts of malicious vehicles. ML can learn the dynamics in the environment and then extract

appropriate features for activity detection. ML strategies can be classified into three groups, namely supervised learning, unsupervised learning (UL), and reinforcement learning (RL). Learning schemes, such as semi-supervised learning, online learning, and learning transfer, can be seen as variants of these three major types. An ML algorithm typically requires two stages: training and testing. In the training state, a system learns from the training data, and in the testing state, predictions are generated to evaluate the model's performance.

### 2.1.5 Different Types of Machine Learning

**Supervised Learning:** Supervised learning algorithms contain a dataset that has both features and labels/targets. The algorithms learn by associating their inputs to the outputs, given a training set of inputs  $x$  and outputs  $y$ . On many occasions, it may be challenging to collect the output automatically [42]. The supervised learning algorithms can observe numerous examples of random vector  $x$  and the associated vector  $y$  and learn to predict  $y$  from  $x$  by estimating the probability of  $p(y | x)$ .

Supervised learning problems can be further divided into classification and regression.

**Unsupervised Learning:** Unlike supervised learning algorithm, UL learns on unlabeled data. This means that the training data only contain input variables, not output variables. These algorithms aim to learn the structure, behavior, or the distribution of the data through modeling and comprehension [43].

**Reinforcement Learning:** RL problem is designed to be an exact representation of the problem of learning from interaction to accomplish a goal [44]. In RL, the learner and decision-makers are known as the agent. The agent can be modeled autonomously to perform optimum sequential actions with or without minimal prior knowledge of the environment, making it incredibly adaptable

and useful in real-time and adversarial contexts. The agents' sequential behavior shows their suitability for CAV security implementations, where threats are becoming increasingly complex, rapid, and pervasive [45].

The environment is the object that the agent interacts with and corresponds to everything outside the agent. The agent's continuous interaction with the environment prompts it to select actions, and the environment responds to those actions by presenting a new situation to the agent. The environment equally gives rise to rewards, which are distinct numerical values that the agent attempts to maximize over time.

The agent and its environment interact in every sequence of discrete-time steps,  $t = 0, 1, 2, 3, \dots$ . In each of the time steps  $t$ , the agent gets some representations of the environment's state,  $S_t \in \mathcal{S}$ ; where,  $\mathcal{S}$  is a set of possible states, which selects actions,  $A_t \in \mathcal{A}(S_t)$ . Here,  $\mathcal{A}(S_t)$  is a set of actions available in state  $S_t$ . A time step later, partly as a consequence of its action, the agent receives a numerical reward,  $R_{t+1} \in \mathcal{R} \subset \mathbb{R}$ , and then finds itself in a new state,  $S_{t+1}$ . Each step leads the agent to implement a mapping from states to probabilities of selecting each potential action. The mapping process is known as agent's policy and is denoted as  $\pi_t$ , where  $\pi_t(a | s)$  is the probability that  $A_t = a$  if  $S_t = s$ . RL mechanisms specify how the agent changes its policy from its experience. Overall, the agent's goal is to maximize the total amount of reward it has over the long run.

## 2.2 Related Work

As vehicles get more interconnected with their external environment, a number of attacks surface and the possibility of leveraging vulnerabilities increases. A growing body of literature has established CAV vulnerabilities and examined the possible effects of successful exploitation of vulnerability while proposing



some mitigating steps [46]. The recent interest on anomaly detection has created a large amount of literature over the past few years. It is challenging in many disciplines, including automotive engineering [47], environmental engineering, and wireless networks [22]. CAV communication can use ML algorithms to detect fault, diagnose, monitor, and intrusion detection [22]. Also, CAV communication can use simple reconfiguration control to prevent or reduce potential loss where anomalies can easily be detected. Various detection mechanisms have been developed in recent years to detect abnormal behaviors and identify their sources [21].

For example, in the field of CAV, recent studies demonstrate the vulnerability of CAV sensors (such as speed, acceleration, and position sensors) to cyberattacks or faults. Sensor behavior with an anomaly is a result of either sensor failure or malicious cyberattack. CAV has many internal and external cyberattack surfaces from which adversaries can act on and exploit [48]–[50]. In [21], the authors conceived a novel model with comprehensive architecture that combines the adaptive extended Kalman filter (AEKF) with a moving vehicle model to detect faults/malicious activities in CAV network. The authors proposed a model that could detect different types of attacks effectively. However, the downside of this approach is that it is greatly affected by uncertainty such as processing noise and, at the same time, is very sensitive to the corruption of outliers [51]. Moreover, Kalman filter-based strategy is computationally complex [52]. [27] shows that fake message intrusion and map network attacks are two of the most dangerous attacks/anomalies in CAVs. For example, fake messages may be communicated by the infrastructure, such as RSUs, or a nearby vehicle. Additionally, malicious vehicles can send fake messages through service advertisements, BSMs, and so on. The fake message communication may put CAV passengers and other road users in life-threatening conditions. In [53], the authors developed an anomaly detection approach using entropy-based approaches in in-vehicle networks. The entropy-based approach has been well

studied in the literature. However, due to traffic variation in CAV systems, the entropy detection technique is vulnerable to a high rate of false positives [54]. In [22], a methodology that can seamlessly detect anomalies and their sources in real-time is developed. The authors created an anomaly detection mechanism by combining DL, particularly convolutional neural network (CNN), with Kalman filter- $X^2$  mechanism to detect and identify anomalous sensor readings in the CAV system. However, the second phase of the model's analysis may not perform well when subjected to false data injection attacks derived statistically due to the independence of statistics characteristics nature of Kalman Filter- $X^2$  [55]. Again, Kalman filter is computationally complex for such anomaly detection [52]. In [19], VANET positional attacks are created using conventional attack methods and a dataset called Veremi (Vehicle Comparison Misbehavior) is developed. The authors developed a detection methodology, namely Maat3, a detection and fusion framework based on subjective logic. The subjective logic framework is deployed on a false position attack. Although the subjective logic utilizes probabilistic models with an explicit notion of uncertainty, reputation and computation here depend on the trust framework structure and often involve discarding information [56]. Again, this method employs a traditional trust technique that goes with a predefined threshold. This technique may not perform well in practice in a real-time scenario, such as in CAV networks [11]. Detection of misbehavior in [57] involves deploying a smart protection system to secure self-driving cars' external communication. The intelligent approach is capable of detecting both gray hole and rushing attacks using intrusion detection systems-based (IDS-based) support vector machine (SVM) and feed-forward neural networks (FFNN). However, this technique relies on selecting kernels and complex computation in the optimization process [58] and considers only a single-attack scenario. [59] addressed the problem of cyber tracking for a platoon that moves in a cohesive form along a single lane and is subjected to different kinds of cyber threats. The authors proposed a

cooperative mechanism that leverages adaptive synchronization mechanisms to mitigate the effects of malicious vehicles. The combined mechanism in the form of closed-loop stability is analytically demonstrated using the Lyapunov–Krasovskii theory. However, the analytical method may not be scalable in real-time scenario [9]. The authors in [16] suggested an intrusion detection approach for user-oriented V2V to protect the network from access denial and false warning generation using Greenshield’s model. Greenshield’s model uses a series of identification rules related to each attack to evaluate the correctness of the information sent by vehicles in a CAV network. A vehicle behavior evaluation technique is used to determine a vehicle’s level of trustworthiness. However, this method may not be scalable in the high dense network with a massive amount of information [11]. In [7], a novel trust method using logistic regression to identify events and malicious vehicles is proposed. In this context, the vehicles iteratively learn about the environment from received messages and then update the value of their neighbors’ trust. A drawback of this model is the complex iterations, which may likely result in detection latency. This thesis addresses and discusses the weaknesses of the above-mentioned cited works. Moreover, it proposes a data-driven anomaly/attack detection mechanism for CAV systems. The proposed approaches as stated in Section 1.5, are used to detect and identify abnormal behaviors. Further, single and multiple anomalies are considered to assess the reliability and robustness of our approach in a realistic network setting.

# Evaluation Methodology for Misbehavior Formulation and Detection

---

The chapter defines the different attack models in CAV and M2M communication and then discusses evaluation strategies and performance metrics used in the study.

### 3.1 Methodology

Misbehavior detection in CAV systems offers a significant advantage since message integrity and correctness are more important than confidentiality in such systems. Misbehavior detection in CAV systems has been tested in vastly different ways with different detection mechanisms. Some of these mechanisms are analytical, simulated, and often limited experimental cases. These evaluation mechanisms are also heavily affected by the type of attacks.

### 3.1.1 Studied Attacker Models

CAV safety applications rely on data exchanged among neighboring vehicles. By inserting fake information into CAV networks, attackers can disrupt the proper functioning of the applications. In particular, attacks can cause applications to signal false warnings or make unnecessary and potentially dangerous maneuvers in a CAV network, which undermines safety [60].

The concept of attack complies with the attack mechanisms discussed in Section 2.2. In this study, we consider the attacks that are more of misbehavior rather than a variety of cryptographic attacks. Driven by the experiments on a real-life message dataset for security applications with CAV, this research considers the usefulness of BSM features and develops normal and anomalous behavior models.

### 3.1.2 CAVs Data Characteristics and Descriptions

The dataset used in this thesis was obtained from the Research Data Exchange (RDE) database of the Safety Pilot Model Deployment (SPDM) program. The SPDM program was implemented with the primary aim of testing CAV in real-life applications and scenarios. The dataset contains detailed and high-frequency data (gathered every 100 ms) collected over two years with participation of more than 2,500 vehicles [22]. The SPDM dataset used in this analysis includes speed (Sensor 1), denoted as speed; lateral acceleration (Sensor 2), indicated as  $A_x$ ; and radius of curvature (Sensor 3), represented as RoC. Such sensor data are used in the attack formulations as well. A dataset of thirty three thousand (33,000) samples are used for both attack formulation and detection process.

As there is no public dataset available for CAVs that includes anomalous behavior in sensor measurements due to attacks and ground truths, we used simulation to produce datasets for our experiments. In particular, we accounted

for three types of anomalous behavior: instant, bias, and gradual drift. We presumed that in sensors, anomalous values exist independently due to either attacks or faults.

Our experiment generated various anomalous dataset rates, denoted as  $\alpha \in \{3\%, 10\%, 50\%\}$ , in which one type of attack or all the three types are equally likely to adversely affect each of the three sensors, as discussed in Section 3.1.3. Explicitly, we sampled a uniform random variable of  $\mathcal{U}(0, c)$  was used at each time epoch (every 100 ms) in the CAV trip to decide whether anomaly exists; if it did, another uniform random variable of  $\mathcal{U}(0, c)$  was used to determine the affected sensor. Depending entirely on the experiment, we randomly sampled from one or three anomalous types with uniform or normal distributions. The generated attacks were added to the base value of the sensor. Algorithm 1 provides the pseudocode depicting the random generation of anomalies.

### 3.1.3 Attack/Anomaly Model for CAV System

- (i) Instant anomaly ( $I$ ): This anomaly was simulated as a Gaussian random variable with 0 and 0.01 representing the mean and variance, respectively, based on an epoch of 200 ms. The mean and variance values were scaled by a constant  $c$ , i.e.,  $c \in \{2000, 4000, 6000, 8000, 10000\}$ , where  $c \times \mathcal{N}(0, 0.01)$  captures the various magnitudes. The corresponding simulation values were added to the sensor's base value.
- (ii) Bias anomaly ( $B$ ): This anomaly was simulated by adding an offset to the observation, which was different from the normal sensor reading. The magnitude of the anomalies was sampled with uniform distribution  $\mathcal{U}(0, c)$ , where  $c \in \{2000, 4000, 6000, 8000, 10000\}$ . In this part of the experiment, we accounted for the different values of anomalous behavior duration  $d$  in the system. The values represent the number of periods the anomalous behavior exists in the system, where  $d \in \{3, 5, 7\}$
- (iii) Gradual drift ( $G$ ): This anomaly was simulated by linearly adding values in decreasing/increasing order to the base sensor values. The linear vector of values was increased from 0 to  $c \in \{2000, 4000, 6000, 8000, 10000\}$  for each experiment. This sequence-generating function was denoted by  $\text{linspace}(0, c)$ . Again, we accounted for the various values of duration  $d \in \{3, 4, 7\}$  of the anomalous behaviors.

---

**Algorithm 1: Connected and Automated Vehicle (CAV) Cyberattack**

---

**Generation Process**

---

```
1  $\alpha$ : rate of anomaly
2  $m$ : number of sensors
3  $D$ : highest anomaly duration
4  $t$ : time epoch
5 for  $t \in T$  do
6   for  $i \in \{1, 2, \dots, m\}$  do
7     if no trace of anomaly occurs at time  $t$  for the  $i$ th sensor then
8       if  $\mathcal{U}(0, c) \leq \alpha$  then
9          $d \leftarrow \text{randi}(D)$ 
10        switch (Choose anomaly type with probability distribution  $f_\omega$ )
11          case Instant:
12            Inject 'instant' anomaly type with parameter  $c_1$ 
13          case Bias:
14            Inject 'bias' anomaly type with parameter  $c_2$  and  $d$ 
15          case Gradual Drift:
16            Inject 'Gradual drift' anomaly type with parameter  $c_3$  and  $d$ 
17          end switch
18        end if
19      end if
20    end for
21  end for
```

---



### 3.1.4 Misbehavior Scenarios and Alert Types

This section explains different emergency alerts in a CAV network. The attack formulations stated in Section 3.1.3 can alter the safety-related BSM features, which eventually can result in false emergency alerts in the CAV network.

#### Emergency Electronic Brake Light (EEBL)

Anomaly in vehicular speed and lateral acceleration ( $A_x$ ) can result in false EEBL notification. An anomalous CAV vehicle  $V_i$  with manipulated speed  $v_i'$ , as shown in Section 3.1.3, can introduce a false reference position. This attack can cause damage such as rear-end collision. In Figure 3.1, an attacker CAV, denoted as  $V_i$ , raises an alert and sends its false location  $(x_i', y_i')$ , marked with a red dotted square with velocity  $v_i'$ , across the network to prevent anyone from detecting its false warning.

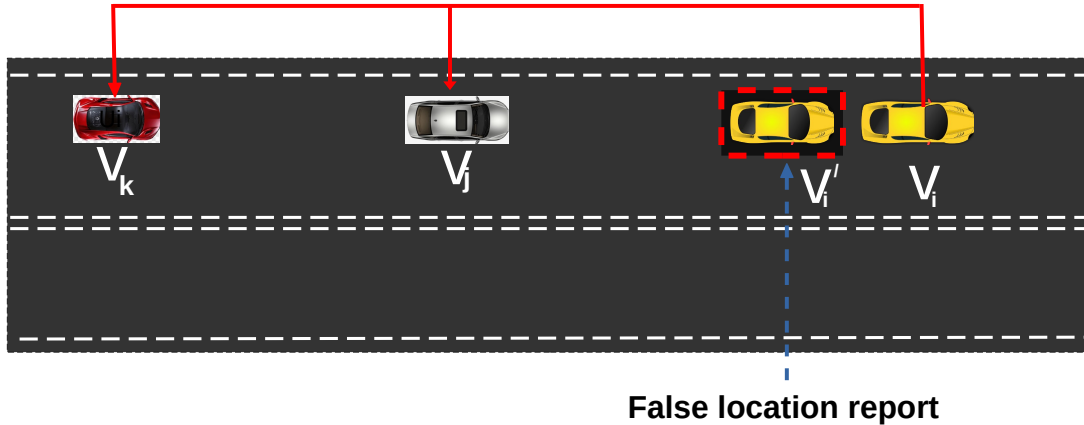


FIGURE 3.1: False emergency electric brake light (EEBL) alert

### Change of Lane (CoL)

CoL alert in CAV networks increases vehicle safety in a dense driving area. CoL devoid of attacks can prevent fatal accidents that can occur when a vehicle unexpectedly switches its current path on a roadway. Figure 3.2 illustrates attacks caused by false CoL alerts. At time  $t$  with velocity  $v_i'$ , attacker  $V_i$  sends a false CoL alert to other CAVs for lane switching with current a false location denoted as  $(x_i', y_i')$ , marked with a red dotted square, instead of its actual position  $(x_i, y_i)$ , as shown in Figure 3.2.

The short distance between  $V_i$  and  $V_k$  prohibits  $V_i$  from changing lanes. However, the false reported location makes the inter-vehicle gap between  $V_i$  and  $V_k$  appear too wide for  $V_i$  to change the lane immediately; here,  $(x_i, y_i) < (x_i', y_i') < (x_j, x_j)$ .

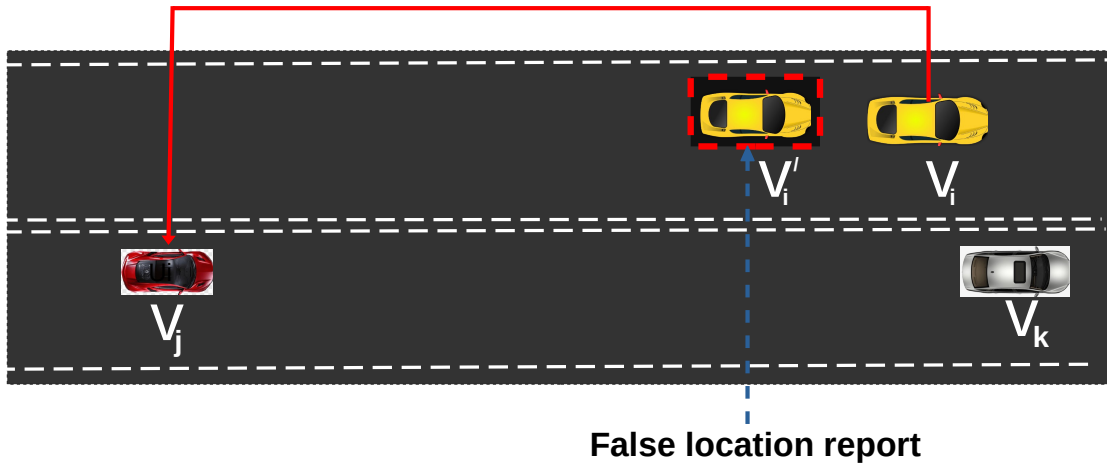


FIGURE 3.2: Change of lane (CoL) attack scenario

### Path Deviation Alert (PDA)

For a straight road, it is estimated that the lateral acceleration should be 0, since the RoC is 0. Conversely, if the road has a non-negative RoC, there is the possibility of this resulting to accident. The lateral acceleration  $A_x$  is related to RoC

by

$$A_x = RoC \times r^2 \quad (3.1)$$

As shown in Figure 3.3, the attacker vehicle  $V_i$  at position  $(x_i, y_i)$  can communicate to vehicle  $V_j$  with a RoC value of 0.  $V_j$  receives a falsified message, adjusts its speed, and keeps heading in a straight line assuming that  $V_i$  is at position  $(x_i', y_i')$ .  $V_j$  will undoubtedly deviate from the lane if it goes by the information given by attacker vehicle  $V_i$ . The vehicle  $V_j$  needs to consider the tangential speed needed to turn the curved road. False alerts like this will invariably lead to vehicle crashes.

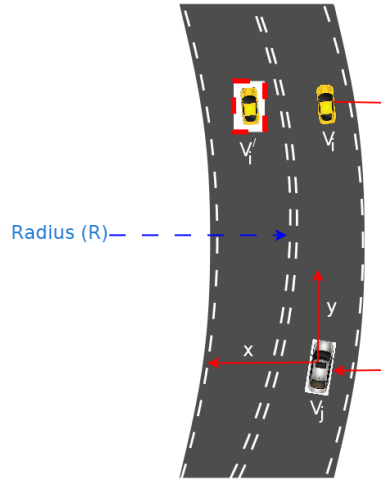


FIGURE 3.3: Path deviation attacker (PDA) Scenario

### 3.1.5 Attacker Model for Machine-to-Machine (M2M) Communication

The synthetic data attack formulation in this context assumed a network of connected vehicles. In the simulation, vehicles were configured to transmit their opinions (BSM features) in a scheduled broadcast. The vehicles were assigned the speed of 24–29 m/s, once every 10 secs. Our proposed model assumes that these vehicles share opinions among themselves in the network and the cloud. When vehicles keep encountering their neighbors, they can form opinions about themselves.

In the attack formulation, we configured malicious and honest code words that convey malicious and honest messages, where  $n$  is the maximum number of messages. The percentage of malicious code word at a given instance is quantified as  $q$ , while the honest percentage is represented by  $100 - q$ .

The  $q$  value ranges from 10 % to 50 % with different probabilities denoted as  $p$ . A malicious code word was formulated by increasing the variance of some fraction of the vehicle's opinion. The simulation results presented a maximum of 50 % of the vehicles to be malicious.

### 3.1.6 Evaluation Strategy

ML-based evaluation approaches allow for a large-scale analysis and evaluation of detector performance. However, only few pieces of research have been done on empirical analysis of CAV attacks. This is because these empirical analyses are challenging to generalize and validate without the extensive implementation of vehicular communication systems. The critical question is that what variables might influence the overall detection efficiency, and such variable should be evaluated individually to make general assumptions about a detection system's suitability.

Variables such as the number of attackers, density of the vehicular network, and the attacks' duration were evaluated in the simulation. By examining these key variables, insights can be obtained on detector performance in different situations and how it responds when a severely attacked message is observed.

Comparatively, the potential to detect attacks in the study of the detection systems is the main focus of this thesis. For optimal results, we expect our method to classify all messages from the attacker as malicious. The robust attack detection approach will give a rough estimate of detector performance, which is important if the attack's aim is uncertain.

### 3.1.7 Misbehavior Detection Metrics

In this subsection, the different metrics that are generally used in the model evaluation are discussed. Performance metrics often differ across publications based on the detector's intent. Perhaps for attack detection in CAV, determining detection mechanisms and selecting the correct one for deployment is tough but important [22]. The following performance metrics are considered in this thesis work:

(i) *Confusion matrix*: The confusion matrix is the simplest method for determining a detector's quality of results. In general, this matrix counts false positive  $fp$ , false negative  $fn$ , true positive  $tp$ , and true negative  $tn$ , as well as their rates for the whole population. These matrices are defined as follows:

- $tp$ : Malicious information correctly labeled as malicious.
- $fp$ : Honest information incorrectly labeled as malicious.
- $tn$ : Honest information correctly labeled as honest
- $fn$ : Malicious information incorrectly labeled as honest.

Metrics derived from this include accuracy,  $F1 - score$ , receiver operating characteristic denoted as (RO), recall, and precision. These derived metrics depend on the type of decisions that a detector can make. In subsequent sections, descriptions of these output metrics are obtained.

(ii) *Accuracy (AR)*: This is the simplest measurement approach used to decide how accurate the classifier is. The accuracy measurement approach can be seen as an intuitive choice for large-scale evaluation. The measurement approach is determined by the number of correct classifications ( $tp + tn$ ). Overall, classifications ( $tp + fp + tn + fn$ ) appear intuitive but suffer from imbalanced sets. Therefore, it is a common practice [18] to

have a two-value quantification, demonstrating the trade-off between increased false positives to minimize false negatives and increased  $fn$  to minimize  $fp$  ones; AR is represented in 3.2 and 3.3, respectively:

$$Accuracy = 100\% \times \frac{\text{Total number of correctly classified process}}{\text{Total number of process}} \quad (3.2)$$

$$AR = \frac{tp + tn}{tp + tn + fp + fn}. \quad (3.3)$$

(iii) Precision and recall (sensitivity): Precision quantifies the importance of the detection events, while recall quantifies which positive rating is observed. For an optimal detection performance, precision and recall must have a value of 1. The degree to which a deviation from these performance values is appropriate depends on the application [18]. The equations for performance and recall are shown in 3.4 and 3.5, respectively:

$$Precision = \frac{tp}{tp + fp} \quad (3.4)$$

$$Recall = \frac{tp}{tp + fn} \quad (3.5)$$

(iv) True positive rate (TPR) and false positive rate (FPR):

*TPR*, also known as sensitivity or recall, is the ratio of the number of positive data points that are correctly predicted as positives. *TPR* [61]:

$$TPR = \frac{tp}{tp + fn} \quad (3.6)$$

Conversely, *FPR* is the proportion of the negative message points mistakenly predicted to be positive to actual negative data points [61]. *FPR* can be mathematically represented as follows:

$$FPR = \frac{fp}{fp + tn}. \quad (3.7)$$

(v) *F1 – Score* : Precision and recall are sometimes combined in weighted mean, know as the *F1 – score*:

$$F1 - score = \left(1 + \beta^2\right) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}. \quad (3.8)$$

The *F1 – score* attains the highest performance level at a value of 1, and this level implies perfect precision and recall. Simultaneously, the worst-case scenario occurs when the precision and recall have values of 0.

(vi) *Area under curve (AUC)*: The RO curve is a coordinate map comprised of a horizontal axis of *FPR* and a vertical axis of *TPR* [62]. RO curve is a comprehensive metric that represents the sensitivity and specificity of a continuous variable. The area under the RO curve is defined as *AUC*. The *AUC* sums up the information in the RO curve with a single number. For an ideal anomaly detector, the *TRP* will be 1 and the *FPR* will be 0, resulting in a step-shaped curve with an *AUC* of 1.

# Misbehavior Detection in CAV Network Based on Machine Learning

---

This chapter discusses different approaches used in this work to accomplish anomaly detection.

### 4.1 M2M Attack Detection Approach

This section discusses the approaches used to detect and identify abnormal behaviors associated with cyberattacks in the M2M network. The approaches are XGBoost, random forest (RandF), and XGBoost optimized with BPSO (BPSO-XGBoost; the proposed method).

#### **Binary Particle Swarm Optimization (BPSO)**

The particle swarm optimization (PSO) algorithm takes its inspiration from the foraging behavior of birds. One intriguing characteristic of PSO is lack of strong



assumption on the criterion, such as continuity or differentiability to find optimal solutions; thus, it can be easily applied to solve optimization problems through simulation of social behaviors [63]. Compared to other optimization methods, the PSO mechanism is robust, scalable, easy to implement, and swift in finding approximate optimal solutions. However, PSO mechanisms are often hard to scale to large problems [64].

For the PSO, the optimization problem's candidate solution is a particle in the hyperparameter space. Each particle has a fitness value, velocity, and position that corresponds to it. The direction and displacement of the particle to search for the candidate solution are determined by the particle's velocity. A fitness value is given to a particle according to its position to describe how good a position is. Thus, finding the best position that has a fitness value becomes the optimization problem. The PSO method finds the optimal solution by looping through a group of initialized random particles.

The number of particles are in the  $D$ -dimensional space is denoted as  $m$ , and the velocity and position of the  $i$ th particle at  $i$ th iteration are represented as  $V_i(t) = (v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t))$  and  $X_i(t) = (x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$ , respectively, where  $i \in \{1, 2, \dots, m\}$ . The particle adjusts its position and velocity according to its current position and velocity, respectively. The best-known position identified by the particle is represented by the particle itself and the entire particle swarm as  $p_{best}$  and  $G_{best}$ , respectively. Each particle will update its velocity and position before the next iteration as given in 4.1:

$$V_i(t) = wV_i(t-1) + c_1r_1 \left( P_i^{best} - X_i(t-1) \right) + c_2r_2 (G_{best} - X_i(t-1)) \quad (4.1)$$

$$X_i(t) = X_i(t-1) + V_i(t), \quad (4.2)$$

Here,  $c_1$  and  $c_2$  represent the particle and population acceleration factors, respectively;  $r_1$  and  $r_2$  represent two independent positive random numbers

between 0 and 1; and  $w$ , as shown in 4.1, denotes the inertia weight. The linearly decreasing strategy of  $w$  is represented in 4.3.

$$\omega = \frac{\omega_{max} + (iter - iter_i) \times (\omega_{max} - \omega_{min})}{iter}, \quad (4.3)$$

where  $iter$  denotes the maximum number of iterations,  $iter_i$  represents the current number of iterations, and  $w_{max}$  and  $w_{min}$  are the maximum and minimum values of  $w$ , respectively. The original PSO is continuous PSO (CPSO), which is implemented and extended to solve several continuous problems [65]. However, PSO has been developed to deal with discrete problems and this PSO is known as BPSO. In BPSO, the position is a binary vector, while the velocity still maintains the continuous vector form. The velocity of BPSO can still be updated with 4.1. However, the velocity entry, unlike in PSO, is used to evaluate the probability that the respective position entry takes a value of 1, as shown in the position update in 4.1.

The transfer function denoted by  $T(V_{ij})$  helps in converting velocities to probabilities; it is expressed as follows:

$$T(V_{ij}) = \frac{1}{1 + e^{-V_{ij}}} \quad (4.3)$$

This transfer function equally helps in updating each bit position as follows:

$$X_{ij} = \begin{cases} 1 & \text{if } \cup(0,1) < T(V_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

The BPSO approach is applied in Section 4.1.1 for optimization process.

### **Extreme Gradient Boosting (XGBoost)**

The XGBoost is a powerful ML algorithm that uses decision tree algorithms as its primary unit. Comparison of XGBoost to the traditional gradient boost

decision tree (GBDT) algorithms, which optimize using the first-order derivative information, shows that the XGBoost has made a significant improvement in optimization. The XGboost performs the second-order Taylor expansion to compute its loss function while reserving and adding the information of the first- and the second-order derivatives, respectively. The retention of information of the first- and the second-order derivatives can make XGBoost approach converge more quickly.

The practical derivative of XGBoost algorithm can be determined as follows: Let the abnormal dataset  $D = \{(x_i, y_i) : i = 1 \dots n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$ , where  $x_i$  is the training set,  $y_i$  is the class label, and  $n$  and  $m$  are the sample data and features, respectively. By setting the maximum iteration time to  $K$ , the integrated approach tree predicts the final result as follows:

$$\hat{y} = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (4.5)$$

Here,  $f_k(\cdot)$  represents a weak learner, and  $\mathcal{F}$  is a set of regression trees.

XGBoost's tree boost algorithm uses Newton boosting rather than gradient boosting to find the best parameters [66] to minimize regularized objective function. The equation for Newton boosting is shown below:

$$\mathcal{L}(\phi) = \sum_i^m l(\hat{y}, y_i) + \sum_k \Omega(f_k) \quad (4.6)$$

Here,  $\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2$  (it represents the complexity of the  $k_{th}$  tree approach);  $m$  means the sample size;  $T$  is the number of leaves in a tree;  $w$  symbolizes the weight of the leaf nodes;  $\gamma$  controls the rate of complexity penalty on  $T$ , and  $\lambda$  controls the rate of regularization of  $f_k$ .

However, the tree ensemble method finds it difficult to minimize loss function in 4.6 with the conventional methods in Euclidean space; thus, the method

is trained using an additive approach as explained in [67]. This approach includes weaker learner  $f_k$ , at  $i - th$  iteration denoted as  $f_t$ , which improves the approach and yields a new loss function as defined in 4.7.

$$\mathcal{L}^t = \sum_{i=1}^n l\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \Omega(f_t) \quad (4.7)$$

Here,  $l$  denotes a training loss function, which measures the difference between the prediction  $\hat{y}_i$ , and the object  $y_i$ ,  $\hat{y}_i^{(t)}$  represents the prediction of the  $i - th$  instance at the  $t - th$  iteration. Furthermore, the second-order Taylor expansion is performed on 4.7 to quickly and conveniently minimize the loss function. This process of Taylor expansion is given by

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^n \left[ l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \quad (4.8)$$

where  $g_i, f_i$  are the first and second derivative, respectively. Now, the  $g$  and  $h$  derivatives can be described as follows:

$$g(i) = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right) \quad (4.9)$$

$$h_i = \partial_{\hat{y}^{(t-1)}}^2 l\left(y_i, \hat{y}_i^{(t-1)}\right) \quad (4.10)$$

The optimal estimate of the weight of each of the leaves in the decision tree (DT) can be formulated as follows:

$$w_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (4.11)$$

From 4.11, the corresponding optimal value for a given node's loss function can be computed as:

$$\mathcal{L}^{(t)} = -\frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (4.12)$$

The details of anomaly detection with XGBoost are presented in Section 4.1.1.

### Random Forest

The RandF classifier is an ensemble method that uses bootstrapping, aggregation, and bagging to train multiple DTs in parallel [68]. Bootstrapping ensures that multiple individual trees are trained in parallel on different subsets of the training dataset using various subsets of available features. RandF classifier aggregates individual DT for the final decision; as a result, RandF classifier has strong generalization [69]. Moreover, bootstrapping maintains the uniqueness of each of the individual DT in the RandF, which helps reduce the variance of the RandF classifier.

The anomaly detection process of RandF can be determined as follows:

Let  $D$  be the anomaly dataset,  $|D| \in \mathbb{N}$  be the number of samples that make up  $D$ , and  $x_i \in X$  ( $0 \leq i \leq |D|$ ) be a sample within the abnormal dataset. Let  $Y$  be the set of class labels for dataset  $D$ ,  $y_i \in Y$  be the label associated with  $x_i$ ,  $\mathcal{C}$  be the RandF classifier,  $|\mathcal{C}| \in \mathbb{N}$  be the number of estimators that compose  $\mathcal{C}$ , and  $tr_j \in \mathcal{C}$  ( $0 \leq j \leq |\mathcal{C}|$ ) be a tree of classifier  $\mathcal{C}$ , which classifies the honest and malicious BSMs.

After training  $\mathcal{C}$  by  $D$  training dataset and  $Y$  class labels, the set of probability labels  $\hat{y}$  is obtained from  $D$  through  $\mathcal{C}$ , as defined in 4.13 below:

$$\hat{y} = \frac{\sum_{j=1}^{|\mathcal{C}|} tr_j^i}{|\mathcal{C}|} \quad (4.13)$$

Here,  $tr_j^i$  represents the output of the tree  $tr_j$  for sample  $x_i$ . The details of the performance results of RandF on M2M malicious activity detection are presented in Section 5.1. Possible estimation is done on the output class  $C_{(0,1)}$ , which comprises of the malicious and honest information in this case. The output label is

given as  $C_{0,1}$ , where 0 and 1 denote malicious and honest information, respectively. The probability of estimation of the output label is expressed as follows:  $P\left(C_{(0,1)} \mid X\right) = \frac{1}{N} \sum_{i=1}^N P_i\left(C_{(0,1)} \mid X\right)$ . An adaptive threshold is automated by the approach while determining the best estimation that can accurately classify both the malicious and honest information.

#### 4.1.1 The Proposed Approach

In this study, the goal is to optimize the parameters of the XGBoost method with BPSO for optimal results in M2M anomaly detection. Although the XGBoost method provides excellent results, the algorithm's training takes longer due to trees built-in sequences. Moreover, the method is hard to tune owing to the enormous amount of parameters [70]. In XGBoost operation, the parameters needed for the anomaly detection are set empirically at maximum iteration  $K$ ; thus, hyperparameter optimization for XGBoost becomes essential at this point [71]. The BPSO maintains excellent results in optimization problems by creating an optimal solution for hyperparameter search.

The task considers six important parameters for tree booster in XGBoost method for optimization: learning rate (*eta*), *max\_child*, *max\_depth*, gamma ( $\gamma$ ), *subsample*, and *colsample\_bytree*. Table 4.1 gives detailed descriptions of the parameters. The BPSO sets to optimize the six parameters of XGBoost that are shown in Table 4.1, and each BPSO's particle represents a six-dimensional vector. Each dimension represents the optimal solution for a single XGBoost parameter. In this task, we further demonstrate the steps for BPSO-XGBoost approach pipeline as follows:

- (i) The dimensions based on the number of parameters to be optimized are identified and particle positions and velocities are randomly initialized. Each particle's position attribute in our task, as shown in 4.18, is a six-dimensional vector with a range that covers the entire search space. Since

TABLE 4.1: XGBoost optimized parameters

Parameter	Default value	Range	Explanation
eta	0.3	[0, 1]	Learning rate used for update to avoid overfitting.
max_depth	6	[0, ∞ ]	A tree's maximum depth. By increasing the maximum depth, the approach becomes more complex and more likely to over-fit.
mini_child_weight	1	[0, ∞ ]	Minimum leaf weight. This is the minimum instance weight for a child
gamma (γ)	0	[0, ∞ ]	Minimum loss reduction needed for further partition on a leaf node of a tree.
subsample	1	(0, ∞ ]	This is the subsample ratio of training instances.
colsample_bytree	1	(0, ∞ ]	This is the subsample ratio of columns while constructing each tree.

the components of each dimension conform to different XGBoost parameters, the initialized range of each dimension remains different. The  $i - th$  particle's position vector with their respective parameters at a time  $t$  is defined as

$$P_{i(t)} = [p_{i(t)}^{eta}, p_{i(t)}^{max\_depth}, p_{i(t)}^{min\_child\_weight}, p_{i(t)}^{gamma}, p_{i(t)}^{subsample}, p_{i(t)}^{colsample\_bytree}] \quad (4.18)$$

Since every particle travels in the same search space, its velocity is initialized to range of (0, 1) in each dimension when  $t = 0$ . The velocity of the  $i - th$  particle at time  $t$  is shown in 4.19.

$$V_{i(t)} = [v_{i(t)}^{eta}, v_{i(t)}^{max\_depth}, v_{i(t)}^{min\_child\_weight}, v_{i(t)}^{gamma}, v_{i(t)}^{subsample}, v_{i(t)}^{colsample\_bytree}] \quad (4.19)$$

Based on the present task, XGBoost is used to detect anomaly problem by

way of classification, the performance of the final training scores as the the fitness function for BPSO is taken. The fitness of the  $i - th$  particle at a given time  $t$  can be expressed as follows:

$$F_i(t) = \left( P_{i(t)} \rightarrow XGBoost \mid_{training\ set} \right)_{metric=training\ score} \quad (4.14)$$

The local optimal value of the global  $m$ -individual particle in time  $t$  is taken as

$$P_{i(t)}^{best} = \max \left( F_{i(j)} \right), 0 \leq j \leq t \quad (4.15)$$

Further, the global optimal value for the global  $m$  individual particle at time  $t$  is represented as

$$G_{best(t)} = \max \left( P_{k(t)}^{best} \right), 1 \leq k \leq m \quad (4.16)$$

1. The position, velocity, and inertia weight obtained from 4.15 and 4.16 for each particle are updated in line with 4.1 and 4.2. Then, a new position is assigned to BPSO-XGBoost method and a new fitness value is calculated in the form of training scores. Comparison of the historical fitness is checked to determine the individual optimal values of the particles to be updated, and it is judged whether global optimal values should be updated. The process iterates until the maximum number of iterations or convergence is reached. The optimal fitness value in the form of the training score (accuracy,  $TPR$ ,  $FPR$ ) and resulting optimal position are then generated. At this stage, BPSO-XGBoost utilizes the optimal parameters to achieve better performance in the detection of anomalies in the M2M communication network. The details of the results of the proposed method are provided in Section 5.1.



### 4.1.2 CAV Attack Detection Mechanism

This section discusses the approaches used to detect and identify abnormal behaviors associated with cyberattacks in the CAV network. The approaches are BDL, one dimensional convolutional neural networks (CNN-1D), support vector machine (SVM), ensemble multilayer perceptron (EMLP), convolutional neural network with attention based long memory (CNN-ALSTM) and Kalman filter convolutional neural network (KF-CNN). The proposed mechanism comprises DWT-BDL and DWT-DDQN.

#### Convolutional Neural Network (CNN)

The receptive field (RF) study provides the theoretical basis of CNN's local perception. CNN consists of layer data, hidden layer, and output layer. The hidden layer includes the convolution, pooling, activation, and fully connected layers. At the core of CNN, the convolution layer is prompted by the RF and computes the data's convolution from the input layer with filters or kernels to extract high-level spatial characteristics. The primary function of the pooling layer is to downsample the number of features. Convolution operations can improve the original features of the data and reduce the noise.

In this thesis, the convolution of the input sequence  $x$  at a time  $t$  is represented as

$$y_k = f \left( \sum_{i=1}^W (w_i \otimes x_{t-k+1}) + b_t \right) \quad (4.17)$$

Here,  $y_k$  is the output feature at time  $t$ ,  $f(x)$  is a nonlinear activation function,  $w_i$  ( $i = 1, 2, \dots, m$ ) is the filter or kernel of length  $W$ , and  $b_t$  is a given offset vector at time  $t$ .

The striking effect of extracting spatial features makes CNN applicable to be used with time-series. Usually, CNN has a two-dimensional kernel for extracting features from an input image. However, CNN's filter can be modified for

time-series data processing. The idea is to allow the filter to pass only in one direction to produce a one-dimensional (1D) output suitable for 1D data input.

The mechanism of CNN-1D approach is applied in real-time detection and identification of abnormal behaviors in a CAV system. A fixed-width sliding window of CNN is inserted on the input data from the sensors' measurements. New observations are gathered from the sensors at each epoch and the sliding window shifts that include the latest observations. Thus, the CNN's input during a CAV trip is a continuous feed of raw sensor data. CNN allows for a holistic view of multiple sensors simultaneously by combining information from other sensors over time. This combination of information helps detect and identify abnormal values. Since the goal is to detect anomalous behavior, each sensor is trained using the labeled sensor readings. If there is an anomalous behavior relating to a given sensor, the response variable is 0 or 1.

Different approaches are trained to detect anomalous behavior from each sensor. A logical OR operator on the approach's outcomes decides whether or not an anomalous behavior has been observed. The CNN parameter values for the architecture in Figure 4.1 are selected based on the series of experiments performed on a validation set to optimize anomaly detection efficiency. This architecture involves three max-pooling and convolution layers. The parameters are a result of several simulations and variations of hyperparameters. For sound performance, the output CNN approach is trained with a random dropout rate of 0.1, and a batch size of 128 is employed. Additionally, batch normalization and rectified linear unit (ReLU) activation functions are used in layers, as shown in 4.1, and Adam optimizer for TensorFlow is implemented in Python for the binary cross-entropy minimization. The following parameters are implemented for the Adam optimization: learning rate,  $\alpha = 0.001$ ; fuzz factor,  $\epsilon = 10^{-8}$ ; and  $\beta_1 = 0.9$ ,  $\beta_2 = 0.99$ . Furthermore, early stopping is introduced to track validation set accuracy with a duration of 200 epochs to reduce the chance of over-fitting.

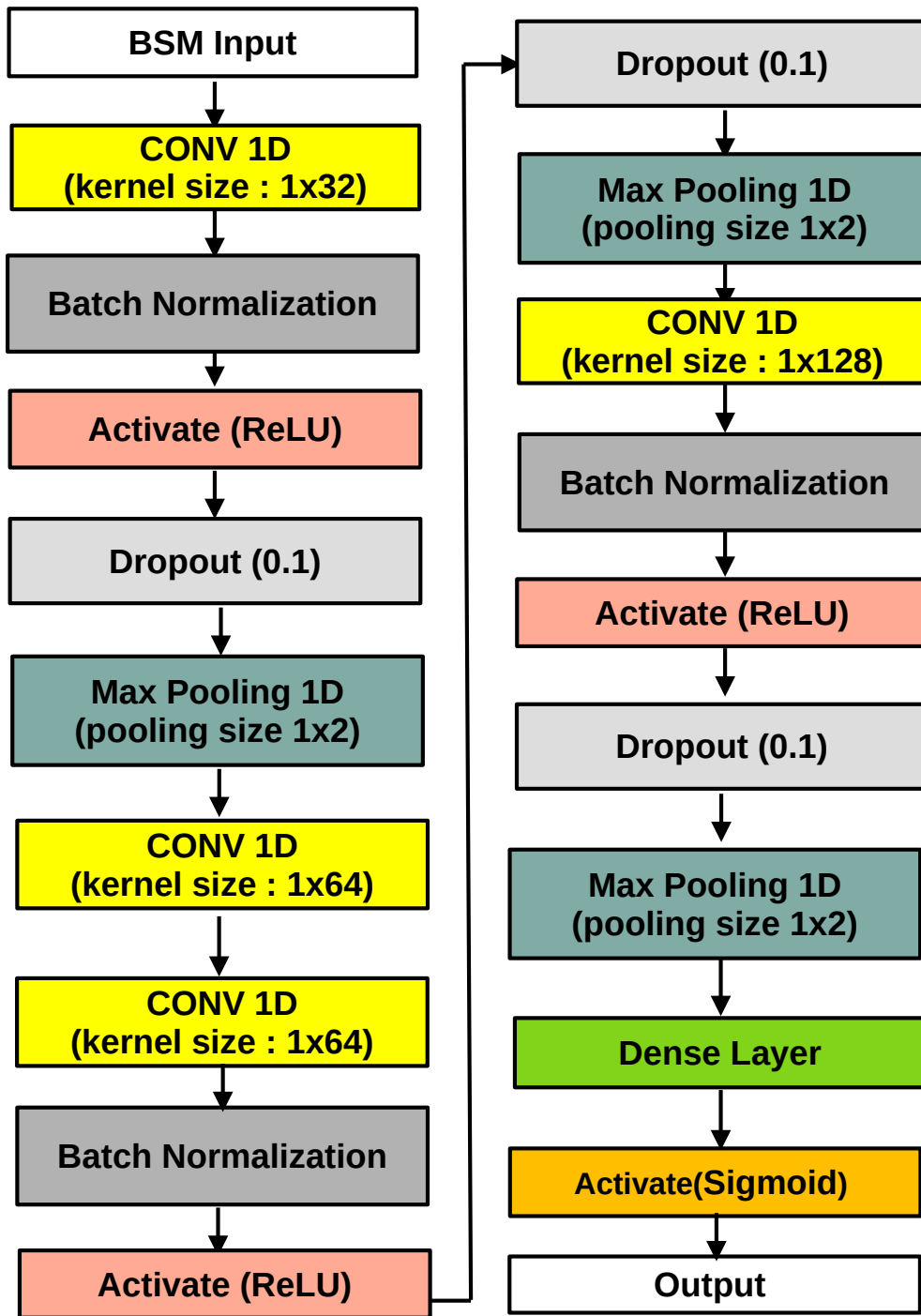


FIGURE 4.1: One-dimensional convolutional neural network (CNN-1D) architecture

The results of this method are presented in Section 5.2.

### Classification Criterion of Convolutional Neural Network (CNN) Algorithm

Consider a time series  $X = (x^1, x^2, \dots, x^m)^T = (x_1, x_2, \dots, x_t) \in \mathbb{R}^{m \times t}$ , given that  $t$  is the time stamp of each value and  $m$  remains the number of features. Here,  $x_t = (x_t^1, x_t^2, \dots, x_t^m) \in \mathbb{R}^m$  is denoted as an input vector at time  $t$ . The time series anomaly detection of the input vector is characterized by two problems: 1) figuring out all the anomaly points and 2) labeling all the target series. CNN approach transforms the anomaly detection mechanism into a classification problem using a sliding window. The sliding window mechanism fragments the entire multi-variable series into continuously shorter sequences and gets the two-dimensional dataset:

$$D = (d_1, d_1, \dots, d_{t-T+1}) = ((x_1, \dots, x_T); (x_1, \dots, x_T), \dots, x_1, \dots, x_T) \in \mathbb{R}^{T \times (t-T+1)} \quad (4.18)$$

Here,  $T$  is the sliding window length and  $d_i$  is known as the anomaly series if it has some anomaly value from the series of origin.

Hence, the problem of detecting anomalies for time series  $x$  is transformed to label each input vector for  $D$ . Training dataset with labels is used to train the CNN approach, and classification approach is applied on the labeled input vector to detect anomalies. The target can be shown as  $\max(\text{Prob}(y_k = 0 | d_j), \text{Prob}(y_k = 1 | d_j))$ , where  $y_k = 1$  denotes honest message and  $y_k = 0$  denotes abnormal message.

#### 4.1.3 Discrete Wavelet Transform (DWT)

DWT transform decomposes the time series data in both the time and frequency domain, even though it is non-stationary. DWT transform has achieved numerous successful applications in engineering fields, such as signal processing and

image processing. The basic idea of a DWT denoising approach is shown below:

$$s(n) = f(n) + \varepsilon(t) \quad (4.19)$$

Here,  $s(n)$  is the observed signal (noisy signal),  $f(n)$  is the real signal, and  $\varepsilon(t)$  represents the Gaussian white noise. The essence of denoising by the DWT is to filter out the  $\varepsilon(t)$  as much as possible.

The theoretical basis of the wavelet threshold denoising method based on Mallat's theory assumes that the low-frequency approximation part and high-frequency information portion of a signal can be fully reconstructed [72]. Suppose an original sensor reading denoted by  $s(n)$  is given by

$$s(n) = \sum_{k \in z} c_{j,k} \varphi_{j,k}(n) + \sum_{i=1}^j \sum_{k=z} d_{i,k} \Psi_{i,k}(n), \quad (4.20)$$

where  $z$  is an integer,  $c_{j,k}$  is the approximate coefficient,  $\varphi_{j,k}$  is the scaling function, while  $j$  is the decomposition level,  $\Psi_{i,k}(n)$  is the wavelet basis function, and  $d_{i,k}$  is the detailed coefficient. Here,  $c_{j,k}$  contains information on the low frequency of the original discrete signal  $s(n)$ , which is stated as follows:

$$c_{j,k} = \langle s(n), \varphi_{j,k}(n) \rangle \quad (4.21)$$

Here,  $\langle s(n), \varphi_{j,k}(n) \rangle$  denotes the orthogonal relationship between  $s(n)$  and  $\varphi_{j,k}(n)$ . The notation  $d_{i,k}$  has the original discrete signal's high-frequency information, which is defined as follows:

$$d_{i,k} = \langle s(n), \Psi_{j,k}(n) \rangle \quad (4.22)$$

Here,  $s(n)$  and  $\Psi_{j,k}$  are orthogonal to each other.

The wavelet threshold denoising approach uses its main profile to be the signal's low-frequency part, while the high-frequency part represents its details.

The details of each level have their noise information after the decomposition of the wavelet. The wavelet threshold function tunes the description coefficients of each level  $d_{i,k}$  and is computed with the approximate coefficients of the last level. The denoising wavelet threshold process is shown in Figure 4.2.



FIGURE 4.2: Wavelet threshold denoising mechanism

In Figure 4.2,  $s(n)$  is the original nosy signal, while  $m_{j,k}$  is the wavelet coefficient obtained as a result of wavelet decomposition of the  $s(n)$ . Here,  $m_{j,k}$  is obtained from the combination of the approximate coefficient  $c_{j,k}$  and the detailed coefficient  $d_{i,k}$ . Further,  $v_{j,k}$  remains the estimated wavelet coefficient after the denoising threshold, and  $\hat{f}(n)$  is the estimated  $s(n)$  derived from the reconstruction of  $v_{j,k}$ . The DWT is applied to improve performances in Sections 4.1.4 and 4.1.7.

### Bayesian Deep Learning (BDL)

BDL framework is required to overcome the challenges in NNs. BDL combines NN's transformation from point to probabilistic estimation by establishing a series of functional transformations in different correlated layers. The NN's transformation is mathematically represented as follows:

$$y_k(x, w) = h \left( \sum_{j=1}^H w_{kj}^{(2)} g \left( \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^1 \right) + w_{k0}^{(2)} \right) \quad (4.23)$$

Here,  $y_k$  is the  $k$ th output of the NN;  $x$  is the vector of the variable  $D$  for the input layer;  $w$  is the combination of the adaptive weight parameters  $w_{ji}^{(1)}$  and  $w_{kj}^{(2)}$  and the biases  $w_{j0}^{(1)}$  and  $w_{k0}^{(2)}$ ; and  $H$  is the number of units in the hidden layer. From the traditional approach, the variable  $\theta$  from the training samples is

estimated by possible minimization of the error function [73], [74].

$$E = E_D + E_W = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^{N_o} \{y_k(x^n; w) - c_k^n\}^2 + \frac{\alpha}{2} \sum_{i=1}^W |w_i^2|$$

Here,  $y_k$  denotes the  $k$ th NN output for  $x^n$ , of the  $n$ th training input data;  $c_k^n$  is the  $n$ th target of the output training data;  $N$  is the corresponding input and output pairs in the target dataset;  $N_o$  is denoted as the number of output variables;  $W$  is the number of parameters in  $w$ ; and  $\alpha$  is the regularization parameter. The variables  $E_D$  and  $E_\theta$  represent the error between the data and the approximation for NN and decay regularization. The NN approach learning process in the Bayesian network is interpreted as a probability. The NN approach within the Bayesian network learning process is to be interpreted in a probabilistic form. The Bayesian phase achieves NN's probabilistic nature by adding strong distribution and uncertainty to the network's weight. The uncertainty in the network model's weight enhances the practical framework in the automatic calculation of error associated with the predictions when dealing with unknown targets. This probabilistic form of the weight also leverages the system to learn from a small amount of evidence [75] when information sparsity is experienced in a given network. Suitable network architecture is then selected, and the model probability is defined as

$$p(w | \mathcal{D}, \alpha, \beta, M) = \frac{p(\mathcal{D} | w, \beta, M) p(\alpha, \beta | M)}{p(\mathcal{D} | M)} \quad (4.24)$$

Here,  $w$  is the adaptive weight parameter;  $\mathcal{D}$  is the data;  $M$  denotes the Bayesian model class that specifies the form of the likelihood function and the prior probability distribution; and  $\alpha, \beta$  are the regularization parameters. At this stage, network training starts with optimizing the input and output data by maximizing the model-specific posterior likelihood by  $w$ . At the end of the preparation, the degree of understanding and generalization is considered adequate. Bayes'

Theorem can be extended as seen below to select the appropriate values for the hyperparameters:

$$p(\alpha, \beta | \mathcal{D}, M) = \frac{p(\mathcal{D} | \alpha, \beta, M) p(\alpha, \beta | M)}{p(\mathcal{D} | M)} \quad (4.25)$$

The hyperparameters  $\alpha$  and  $\beta$  are assumed to be known. Initial values for  $\alpha$  and  $\beta$  are chosen as seen in Figure 4.3, and the associated values of  $w$  are obtained by maximizing their posterior likelihood. Using the following relationship, the hyperparameters are re-estimated, where their MAP values are based on uniform prior values for  $\alpha$  and  $\beta$ , and the estimate of these values maximize evidence  $p(\mathcal{D} | \alpha, \beta, M)$  in 4.25. The estimated values of  $\alpha$  and  $\beta$  are represented as

$$\alpha' = \frac{\gamma}{2E_D} \quad (4.26)$$

$$\beta' = \frac{N - \gamma}{2E_D} \quad (4.27)$$

The  $\gamma$  parameter calculates the approximate number of parameters whose values, rather than the prior values, are controlled by the data, i.e., the number of well-determined parameters.

The Bayesian approach achieves the correct solution by allowing objective comparison among different models. The most probable model class within a set of classes  $M$  of  $N_m$  (no of candidates) is obtained in Bayesian sample selection by applying the Bayes theorem as follows:

$$p(M_j | \mathcal{D}, \mathcal{M}) \propto p(\mathcal{D} | M_j) p(M_j | \mathcal{M}) \quad (4.28)$$

The factor  $p(M_j | \mathcal{D}, \mathcal{M})$  is known as the evidence provided by data  $\mathcal{D}$  for the model class  $M_j$ . The user's judgment on the initial plausibility of each NN approach is expressed by the prior probability  $p(M_j | \mathcal{D}, \mathcal{M})$  over the set of model classes  $M_j$  for  $j=1, \dots, N_m$ , where



$$\sum_{j=1}^{N_m} p(M_j | \mathcal{M}) = 1 \quad (4.29)$$

The last problem to be discussed when deciding the optimum architecture is the relative value of each input variable is the automatic relevance determination (ARD). Using real-system data, distinguishing the important variables from the redundant ones may be difficult. However, in the Bayesian approach, the ARD method proposed in [76] can address this problem.

In the method, the input variable is associated with a separate hyperparameter  $\alpha$ , representing the inverse variance of that input parameter's prior distribution. In this way, each hyperparameter reflects the importance of input: A small value implies that a considerable weight parameter value is permitted and the resulting input is important; on the contrary, a significant weight parameter value  $\alpha$  is allowed, and the associated weight parameter confines to zero, and thus the corresponding input is less relevant [76], [77].

The ARD minimizes evidence for the class of model by identifying the hyperparameters in BDL architecture. High hyperparameter values are removed from the approach at this minimisation stage, and an equilibrium design is re-estimated for a new implementation.

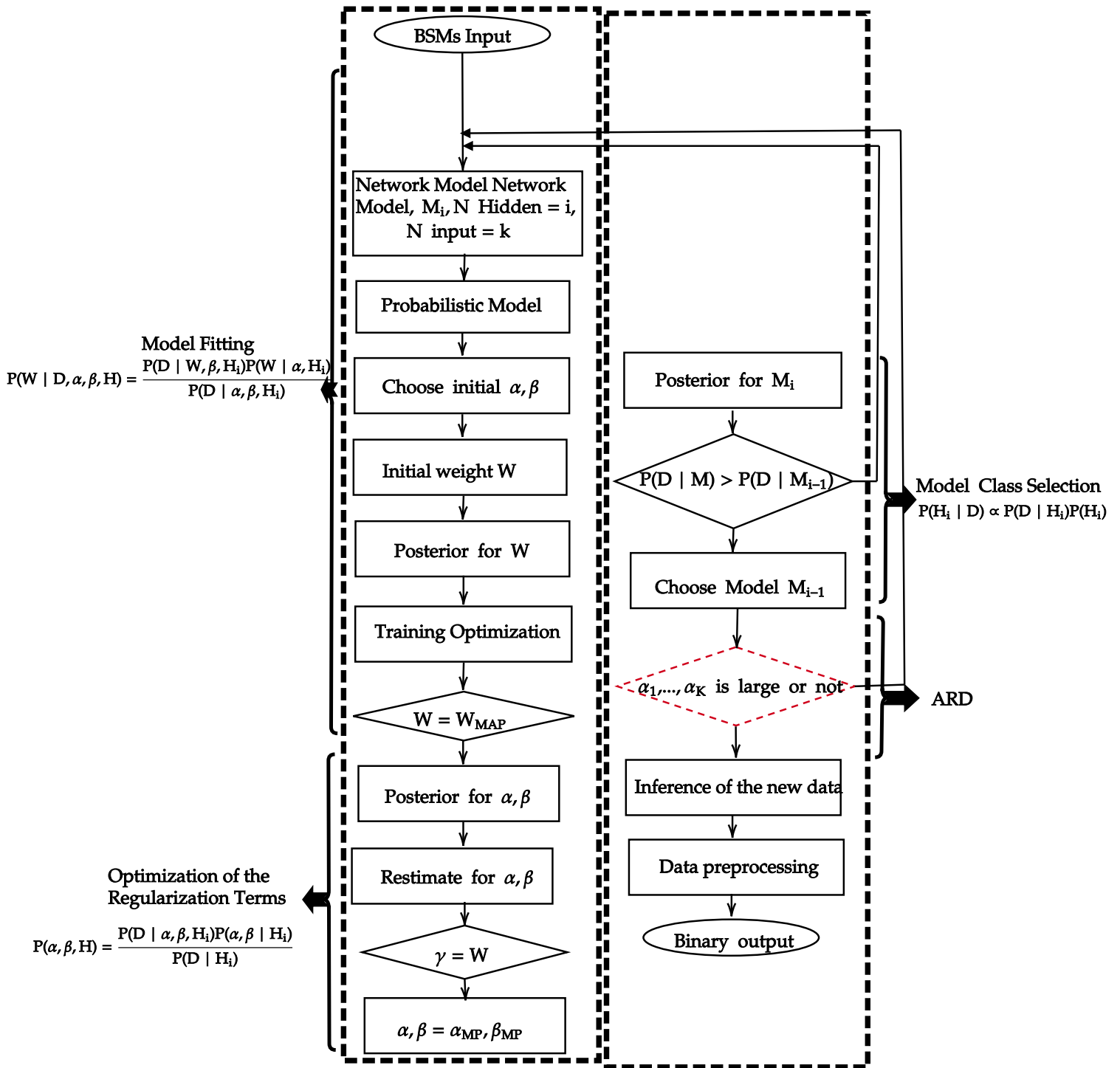


FIGURE 4.3: Bayesian hierarchical framework for neural network

#### 4.1.4 Discrete Wavelet-Based Deep Reinforcement Learning with Double Q Learning (DWT-DDQN)

(i) Dataset preparation

In this case, the anomaly problem is formulated as an RL problem. RL is an autonomous agent that interacts with the environment, takes action, receives a reward from the environment, and learns to predict anomalies with high accuracy. The formulation is achieved by replacing the environment with the mini-batch sampling process of the recorded training BSM anomaly dataset. The sampling process generates sets of training iterations that construct pseudo-environments for anomaly detection. The BSM dataset includes  $N$  samples of BSM features and related labels with several possible binary values. The mini-batch samples are further assimilated into a DRL concept by considering the features as state  $s$  and the labels values as action  $a$ . We trained with mini-batches of BSM samples consisting of (1) a state, (2) the right label, and (3) a corresponding state. A mini-batch is a randomly chosen subset of samples from the BSM dataset. Each training pass uses a separate mini-batch, updated using random BSM dataset sampling from the simulation. The mini-batch configuration used for the proposed approach detection analysis is presented in Figure 4.5. In this case, a mini-batch is made up of  $(n + 1)$  random dataset samples. The process of generating each mini-batch takes a random permutation of the dataset before the process is initiated. Then,  $n+1$  random dataset samples are selected, and the consecutive samples start from a random index  $(t)$ .

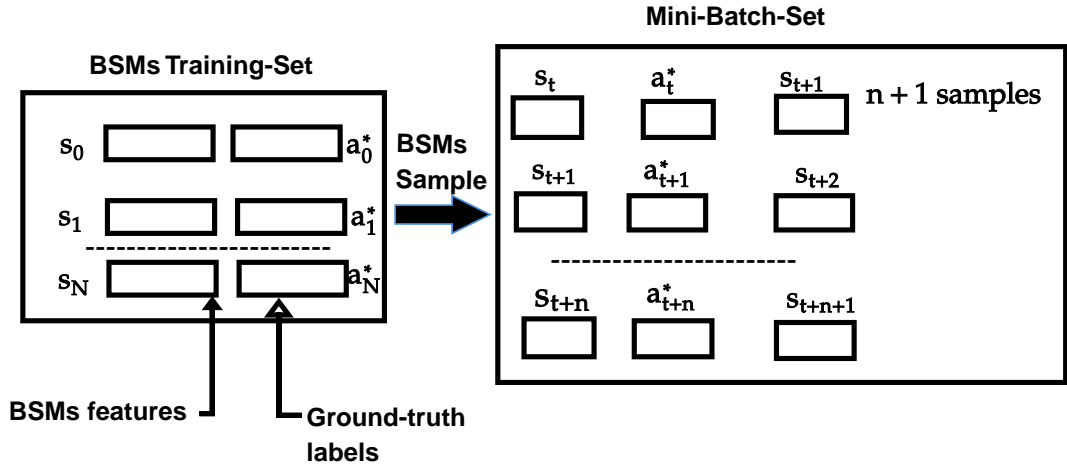


FIGURE 4.4: Basic safety message (BSM) dataset preparation for the training of the double deep Q-network (DDQN) approaches  
Source: Adapted from [78]

We have  $S_{t+1}$  and  $a_{t+1}^*$  as the next state and the action, respectively.

(ii) DDQN approach description

Consider a target Q-value [79] represented as

$Q^*(s, a) = \mathbb{E}_{S_{t+1} \sim \epsilon} [r + \gamma \max_{a_{t+1}} Q^*(S_{t+1}, a_{t+1}) \mid s, a]$ , where  $r$  is the reward,  $\gamma$  represents the discount factor, and  $a$  is the action. Taking the maximum estimated value,  $\max_{a_{t+1}} Q^*(S_{t+1}, a_{t+1})$ , is equivalent to implicitly taking the maximum overestimate value. The systematic overestimation can result in maximizing the bias in learning [79]. This type of Q-learning goes with the bootstrapping learning estimate from an estimate, which can be problematic.

To deal with this concern of overestimation, DDQN for model structure is selected. The structure involves using two different Q-value estimators, each of which updates the other. Independent estimator helps to unbiased Q-value estimates of the action chosen using the oppose estimator [79]. Thus, maximizing bias is avoided by disentangling our updates from bias estimates.

(iii) DWT-DDQN anomaly detection strategies

A Q-function provides the highest expected reward based on a specific state and action. The extent of reward depends on the state and action pair,  $Q(s | a)$ . The Q function assists in executing the policy function. In policy function evaluation, each state's required plan of action, which varies by state and action. Figure 4.5 demonstrates the development of a generic BSM sample made by the current state ( $s_t$ ), the ground truth label for the current state ( $a^*$ ), and the next step ( $s_{t+1}$ ).

The DDQN network comprises three layers, each with ReLU activation to ensure positive Q function. The network is trained with binary cross-entropy between the Q-value estimate of the NN for the current state ( $q_t$ ) and the reference value ( $q_{ref}$ ). The  $q_{ref}$  is derived by the current reward and the next state Q-value denoted as  $q_{t+1}$  and the discount factor  $\lambda$ . The reward is a binary output denoted by 1 or 0, which depicts correct or incorrect prediction  $\hat{a}_t$ . The value of the ground truth label for the current state is represented by  $a_t^*$ , and the predicted values are represented by  $\hat{a}$ . If the current and predicted state values are equal, the rewards are 1; otherwise, they are 0. The predicted value of the current state is obtained by iterating the Q function with the current ( $st$ ) and the values of the labels  $\{\{a\}\}$ . The iteration is represented as  $(Q(s_t, \{\{a\}\}))$ , while a and p are possible actions and cardinalities.

The maximum action value  $arg_{amax}(Q(s_t, \{\{a\}\}))$  generated from the iteration is applied to E-greedy algorithm. The algorithm selects the probability value  $p$  or a random action with probability action  $1-p$ , and this selection step provides the predicted action ( $\hat{a}$ ).

The Q-value for the next state ( $\hat{q} + 1$ ) is determined using the target Q function. The goal of this extra Q function (target Q function) is to prevent the moving target effect when doing gradient descent over  $(\hat{q}_t - q_{ref})^2$

and the  $q_{ref}$ 's recursive dependency on the training network.

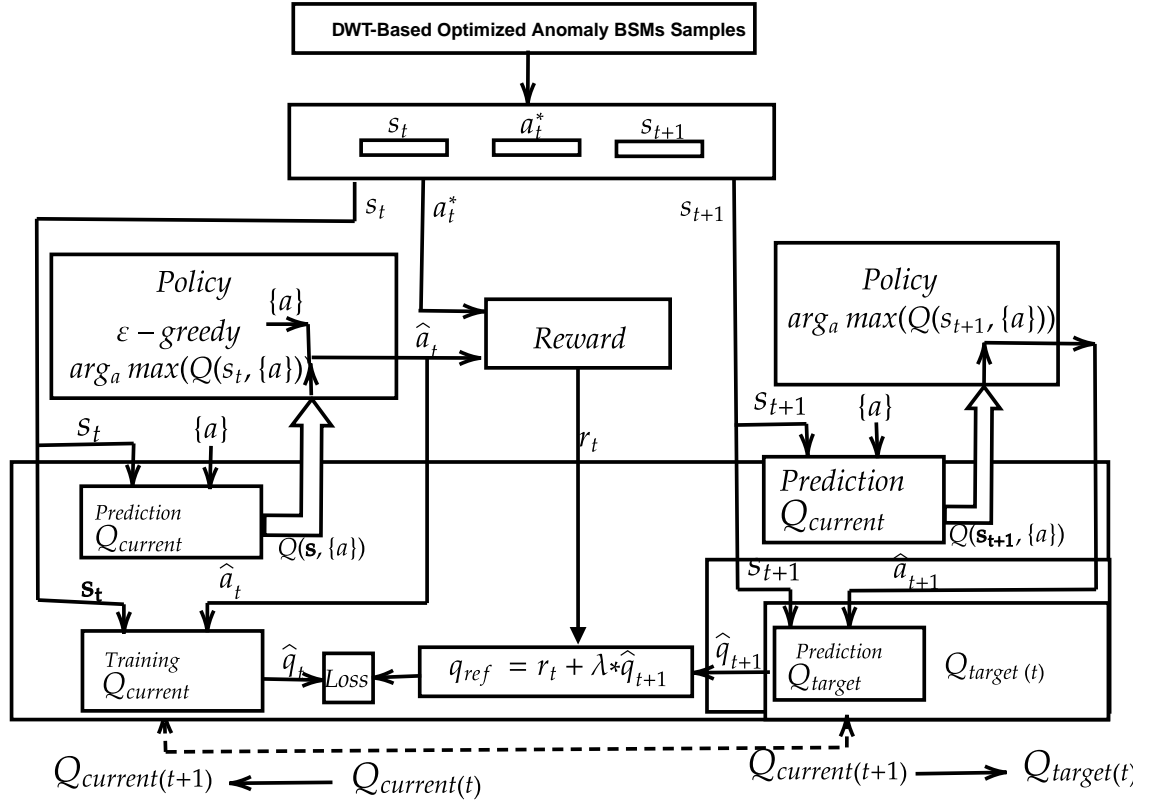


FIGURE 4.5: Double deep Q network (DDQN) approach training scheme

Source: Adapted from [78]

The DWT-DDQN parameter values are selected based on experiments performed on a validation set to optimize anomaly detection and identification efficiency. The architecture of the proposed DWT-DDQN approach involves fully connected NN comprising three layers, with ReLU activation for all layers, including the last one to ensure a positive Q-value. DWT-DDQN method is trained with a random dropout rate of 0.1, and a batch size of 128 is employed. Additionally, batch normalization and ReLU activation functions are used in layers, and Adam optimizer for TensorFlow is implemented in Python for the binary cross-entropy minimization. Early stopping is introduced to track validation set accuracy with a duration of 200 epochs to reduce the chance of

over-fitting. The performance of DWT-DDQN method is discussed in Section 5.2.4.

#### 4.1.5 Support Vector Machine (SVM)

Consider a dataset  $\{x_1, x_2, \dots, x_N\}$ , where each  $x_i \in \mathbb{R}^N$  represents a member of the class of interest. The SVM's goal is to find a hyperplane with the maximum value that separates the outlier from the rest of the results in the dataset. We can define a mapping function  $\phi$  as a kernel mapping function  $N \rightarrow \mathcal{F}$ . To differentiate the hyperplane, SVM solves the quadratic programming problem shown below:

$$\underset{w \in \mathcal{F}, \xi, \rho}{\text{Minimize}} \frac{1}{2} \|w\|^2 + \frac{1}{vN} \sum_{i=1}^N \xi_i - \rho \quad (4.30)$$

The problem is subject to  $(w, \Phi(x_i)) \geq \rho - \xi, \xi \geq 0 \forall i = 1 \dots N$ ,

where  $w$  is a vector perpendicular to the hyperplane in  $\mathcal{F}$ ,  $\rho$  is the distance to the origin, and  $N$  is the number of data points. A range of slack variables  $\xi_i \geq 0$  is added to allow outliers to lie on the margin. The constant parameter  $v$  represents the decision boundary's *FPR* that classifies normal and abnormal sensor readings in a range of  $(0, 1)$ . The decision variables  $w$  define the most generalizable linear decision boundary in an infinite-dimensional space to decide a region in the input space encompassing at least  $1 - v$  percentage of the data points.

The decision/slack variable  $\xi$  is used to penalize the degree of violation of the constraints  $((w, \Phi(x_i)) \geq \rho)$ .

The result of the performance of SVM is shown in Section 5.2.4.

#### 4.1.6 Ensemble Multi layer Perception (EMLP)

An EMLP method comprises a series of layers, each of which has neurons attached to the next layer to create a unidirectional feed-forward structure.

The method has m-layer that figures the dataset input-output pairs  $(\vec{x}_i, y_i)$ , where  $y_i$  represents the 1D output  $y_i \in \{0, 1\}$ , where 0 and 1 denote abnormal and normal sensor reading, respectively, on n-dimensional BSM input vector  $\vec{x} = \{x_1, x_2, \dots, x_n\}$ . The dataset input and output pair size is denoted as  $X = \{(\vec{x}_1, y), \dots, (\vec{x}_N, y_N)\}$ . For each hidden layer from  $l_1$  to  $l_{m-1}$ , the weight sums and outputs are computed as follows:

$$\begin{aligned} s_i^k &= \overrightarrow{w}_i^k \cdot \overrightarrow{z}^{k-1} + b_i^k \\ s_i^k &= b_i^k + \sum_{j=1}^{r_{k-1}} w_{ji}^k \cdot z_j^{k-1} \quad \text{for } i = 1, \dots, r_k \end{aligned} \quad (2)$$

Here,  $w_{ji}^k$  is denoted as the weight of the link between  $i_{th}$  neuron of  $l_k$  layer and  $j_{th}$  neuron of  $l_{k-1}$  layer;  $\vec{x}$  represents the input vector to the approach. The  $b_i^k$  represents the bias for  $i$  neuron in the layer  $l_k$ , and  $r_k$  represents the number of neurons in layer  $l_k$ ; further,  $s_i^k$  denotes the product of the summation with the bias of neurons  $i$  in layer  $l_k$ .

The output neuron  $i$  in layer  $l_k$  is denoted as  $z_i^k$  and can be computed as follows:

$$z_i^k = f_h(s_i^k) = \begin{cases} s_i^k, & s_i^k > 0 \\ 0, & \text{else} \end{cases} \quad \text{for } i = 1, \dots, r_k \quad (3)$$

Here,  $f_h$  is the activation function at hidden layer. Generally, for MLP, the output  $z$  is obtained by the feed-forward mechanism. The computing output  $\hat{y}$  that computes the output sensor reading into normal and abnormal value is shown as

$$s_1^m = \overrightarrow{w}_1^m \cdot \overrightarrow{z}^{m-1} + b_1^m = b_1^m + \sum_{j=1}^{r_{m-1}} w_{j1}^m \cdot z_j^{m-1} \quad (4)$$

$$\hat{y} = z_1^m = f_z(s_1^m) = \frac{1}{1 + e^{-s_1^m}} \quad (5)$$



The EMLP parameter values are selected based on the experiments performed on a validation set to optimize anomaly detection and identification efficiency. This EMLP architecture involves two stacks of three hidden layers and is trained with a random dropout rate of 0.1, and a batch size of 128 is employed. Additionally, batch normalization and ReLU activation functions are used in layers, and Adam optimizer for TensorFlow is implemented in Python for the binary cross-entropy minimization. The following parameters are implemented for the Adam optimization: learning rate,  $\alpha = 0.001$ , batch size of 128, dropout rate of 0.1 and ReLU activation. Furthermore, early stopping is introduced to track validation set accuracy with a duration of 200 epochs to reduce the chance of over-fitting. The performance obtained by using EMLP is detailed in Section 5.2.4.

#### **4.1.7 Bayesian Deep Learning-Empowered Discrete Wavelet Transform (DWT-BDL)**

The detection efficiency of the BDL approach is improved by proposing a new framework (DWT-BDL) based on the reliance of DWT and BDL, as shown in Figure 4.6. Prior to feeding the data into the BDL detection algorithm, DWT is added to the BSM sensory information for denoising (as explained in section 4.1.3). The noisy sensory BSM data are decomposed by transforming them into an orthogonal domain, followed by processing operations on the resulting coefficients. Eventually, through the reconstruction process, the sensory input is transformed back to its original state.

The denoised reconstructed BSM sensory input is fed into the BDL algorithm for further examination and anomaly detection, as explained in Section 4.1.3. This stage of anomaly detection is achieved by first splitting the data into training and testing datasets. The proposed approach is trained to develop a

prediction on the training dataset, while the testing dataset is fed into the algorithm for prediction test.

To further improve the detection and identification efficiency of the BDL algorithm, a new framework is developed based on the reliance of BDL and DWT, as shown in Figure 4.6. In this approach, before feeding the data into the BDL detection algorithm, DWT is added to the BSM sensory information for denoising. The noisy sensory BSM data are decomposed into an orthogonal domain, followed by processing operations on the resulting coefficients. Eventually, through the reconstruction process, the sensory input is transformed back to its original state based on the criteria discussed in Section 4.2.

The denoised reconstructed BSM sensory input is fed into the BDL algorithm for further examination and anomaly detection. The performance results of the proposed approach are shown in Section .

### **Classification Criterion of the Proposed Approach**

This section provides a detailed description of the proposed method's anomaly detection in the CAV network, and Figure 4.6 gives a detailed description of the pictorial representation. The value  $\vec{x}$  is a vector of BSM sensory input of  $D$  variables. The  $\vec{x}$  is a piece of evidence to be predicted. Moreover,  $c_k$  and  $M$  are the relevant class outputs (ground truths) estimated by the proposed approach. A value of 0 or 1 is assigned to  $\vec{c}_k$  variable, where 0 and 1 represent the normal and anomalous information, respectively. Here,  $\vec{c}_k \in \{anomalous, normal\} \equiv \vec{c}_k \in \{0, 1\}$  considering a binary classification.

The mathematical representation of the classification process can be expressed using Bayes Theorem in C.2:

$$p(C = \vec{c}_k | M_j, D = \vec{x}_i) = \frac{p(M_j | C = \vec{c}_k, D = \vec{x}_i) p(C = \vec{c}_k | D = \vec{x}_i)}{p(M_j | D = \vec{x}_i)} \quad (4.31)$$

By application of total probability theorem, C.2 can be represented as shown in 4.32.

$$p(C = \vec{c}_k | M_j, D = \vec{x}_i) = \frac{P(M_j | C = \vec{c}_k, D = \vec{x}_i)p(C = \vec{c}_k | D = \vec{x}_i)}{\sum_{c \in (\text{anomalous}, \text{normal})} [p(M_j | C = c_k, D = \vec{x}_i)p(C = c_k | D = \vec{x}_i)]} \quad (4.32)$$

From 4.32, it is assumed that the individual reports remain independent [10]. From 4.33, the conditional probability of normal and anomalous information in the CAV network is deduced. From the conditional probability of normal and anomalous information in vehicular networks, the following mathematical expression is further deduced:

$$p(\text{anomalous} | \vec{x}) + p(\text{normal} | \vec{x}) = 1 \quad (4.33)$$

from 4.33, it can be inferred that  $\vec{X}$  is malicious when  $\vec{X} = 1 - p(\text{hon} | \vec{x})$ .

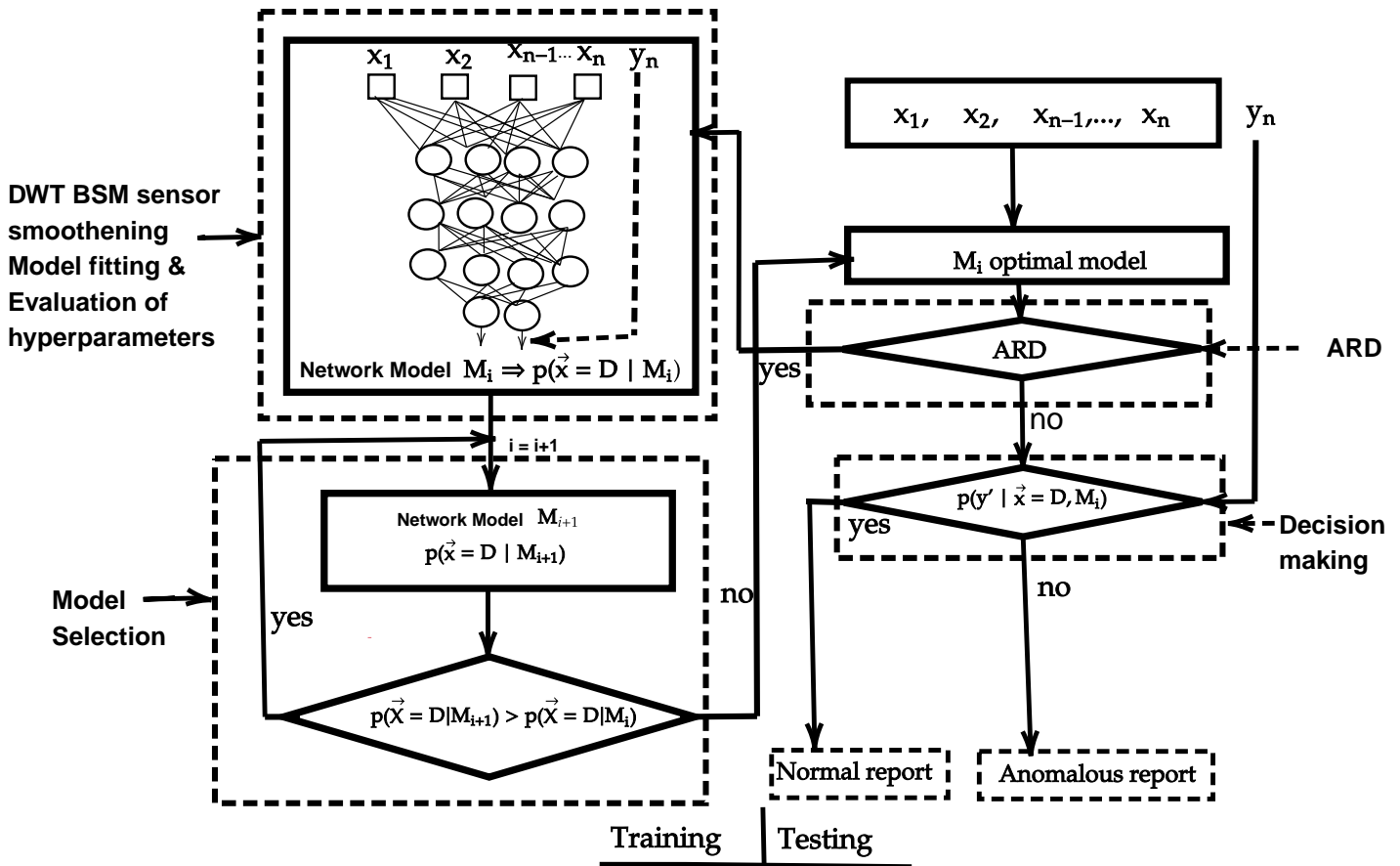


FIGURE 4.6: Bayesian hierarchical framework for neural network

The values of BDL parameters are selected based on the series of experiments performed on a validation set to optimize anomaly detection and identification efficiency. In this case, the BDL architecture is assumed to comprise of TensorFlow probability (TFP) of four hidden layers, with 20 nodes and one bias in the first layer and ten nodes and one bias each in the rest of the layers. ReLU activation function is applied to train the network, while Adam optimizer with default learning rate minimizes the validation set binary cross-entropy loss. The details of the performance of BDL are presented in Section 5.2.

# Results and Discussion

---

## 5.1 Results: Machine-to-Machine (M2M) Model Detection Performance

This section presents the simulation results of the proposed and state-of-the-art (SOTA) ensemble approaches, namely XGBoost and RandF. Their performances are compared for attacking vehicle message densities. The performance metrics such as *AR*, *TPR*, and *FPR* are used to demonstrate the proposed model's behavior and its superiority over the other models.

Figure 5.1 provides the results of the proposed and SOTA approaches. It can be seen that the proposed BPSO-XGBoost approach improves the detection performance of both XGBoost and RandF approaches in all the scenarios of attack density. For instance, at attack density of 10 %, the accuracy of BPSO-XGBoost approach increases by 0.02 and 0.04 over RandF and XGBoost, respectively. The proposed approach displays superior performance in accuracy over the SOTA since the combined features of the proposed approach strengthen its detection capability. The XGboost works with optimized parameters as a result of the adaptive search ability of BPSO. As it is observed in Figure 5.1, generally

the detection approaches show increased performance with high magnitude of attacker vehicles. Here, the increase in accuracy is attributed to the fact that a high attacker vehicle percentage in a network generates larger distribution, which can deviate from the true values of the normal sensors' behaviors. The larger the anomaly distribution, the easier for the detection approaches to extract meaningful information in detecting anomaly present in the network.

The *TPR* performance of the approaches with different attacker densities is simulated and results are presented in Figure 5.2. Results show that XGBoost and RandF performed similarly in all the attack densities. However, the proposed approach consistently performed better than the XGBoost and RandF and approached almost 100 % *TPR* value when the attack vehicles are increased in the simulations. The proposed approach results in high detection accuracy for the true event. For instance, at 10 % of attack density, the *TPR* performance difference between the proposed approach and the other approaches is around 5 to 7 %.

The proposed approach's strength compared to the conventional approaches is further demonstrated with *FPR* metric. Figure 5.3 indicates that in all the attack density scenarios, the proposed approach exhibits a very low misclassification rate of almost 2 % compared with the selected baseline approaches. The performance improvement demonstrates the effectiveness of BPSO in XGBoost parameter optimization, which strengthens the proposed approach's detection capacity.

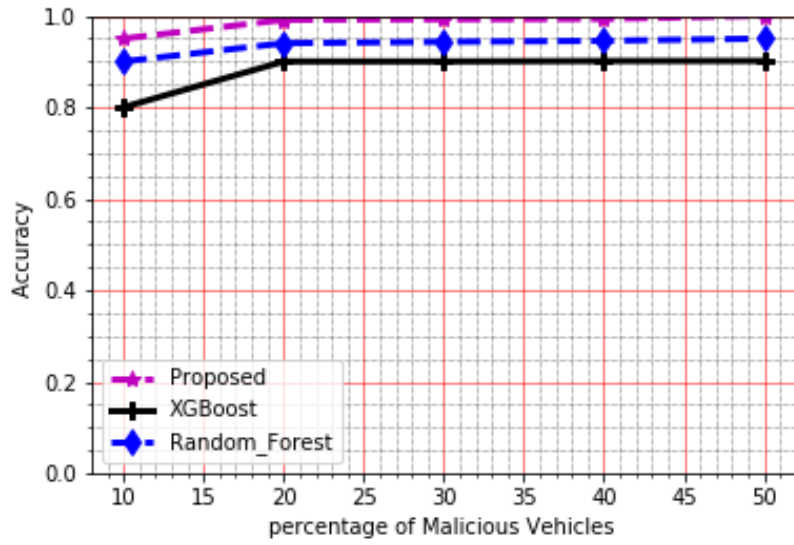


FIGURE 5.1: Accuracy vs. attacker percentage scenario

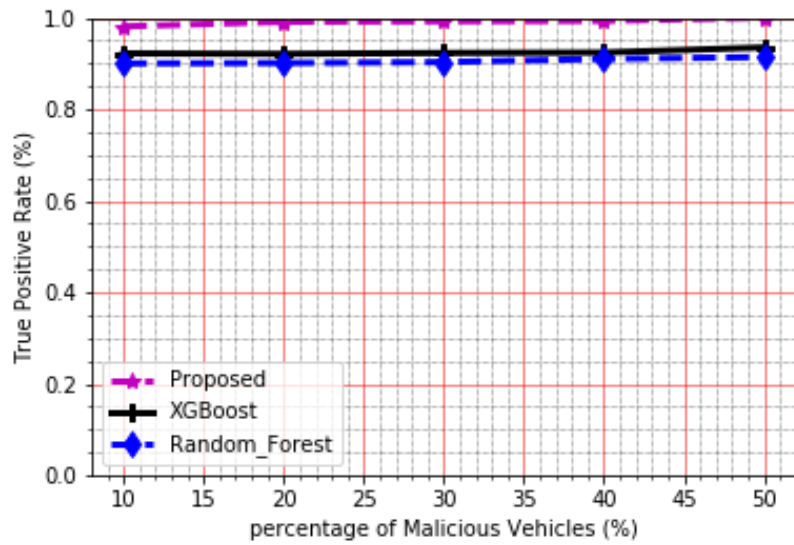


FIGURE 5.2: True positive rate vs. attacker densities

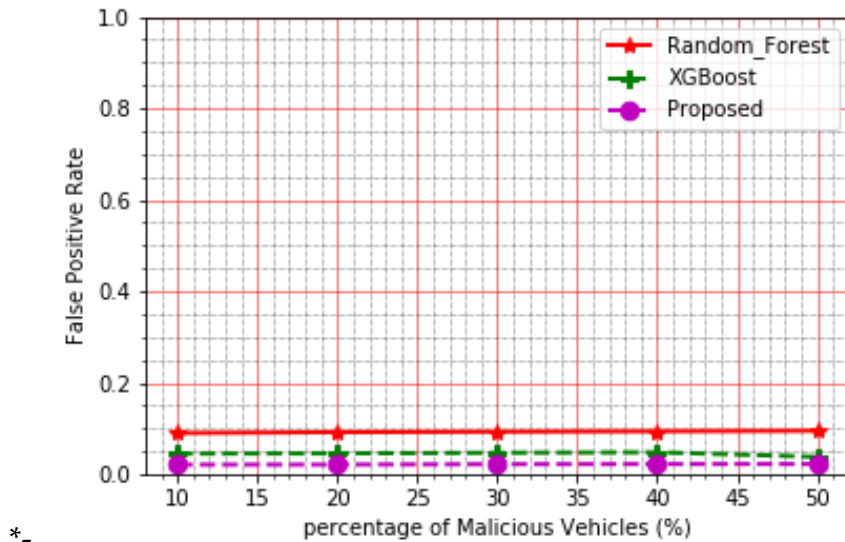


FIGURE 5.3: False positive rate vz. attacker densities

### Discrete Wavelet Transform (DWT) Pre-Analysis of the Data

BSM variables are usually in time series format, and successive time series values are not necessarily independent but are strongly correlated. This makes it difficult to establish successful feature selection strategies for the functions that operate directly on time-series data. To mitigate the problem, DWT, which comes with the flavor of both a feature extractor and denoising techniques, as detailed in Section 4.1.3, can be applied to time series data to transform them from a time domain to a frequency domain.

In this section, we mainly focus on the denoising of the BSM attributes with DWT. The BSM data have some disturbances due to the noise from the measurement and the estimated error, which may cause problems in the training phase of the anomaly detection approach. In addition, the inbuilt noise in the system can induce an outlier effect, which can impact the performance of the attack detection approaches. The details of the operation of DWT are examined on the selected BSM sensors and the simulation for the experiment is carried out with Python.



The BSM data are denoised to make them smoother and accelerate the convergence of the loss function during the training process of the proposed detection approach. Table 5.1 demonstrates the abnormal and denoised readings of BSM attributes. On carrying out our simulations at  $d = 3$  in various network densities,  $c$ , results indicate that DWT presents finer coefficients of the noise measurement with a decrease in standard deviation in all the three anomaly cases. The effects of combining filtering and denoising process of DWT with the proposed approach is shown in Section 5.2.

TABLE 5.1: Descriptive statistics of selected basic safety message (BSM) variables

Instant anomaly (Network size m)	$\mu$ (Anomaly)	$\sigma$ (Anomaly)	$\mu$ (Wavelet dB[12])	$\sigma$ (Wavelet dB[12])
2000	10.307401	6.170231	10.300251	5.7634621
4000	10.307407	6.170261	10.300258	5.763563
6000	10.307414	6.170258	10.300262	5.7635907
8000	10.307417	6.1702566	10.300263	5.7635937
10,000	10.307415	6.170257	10.300264	5.7635903
Bias Anomaly (Network Size m)	$\mu$ (Anomaly)	$\sigma$ (Anomaly)	$\mu$ (Wavelet dB[12])	$\sigma$ (Wavelet dB[12])
2000	0.5829332	1.5610949	0.5813745	1.1946311
4000	0.53638434	1.581967	0.53705674	1.0952997
6000	0.6300388	1.7301117	0.6321792	1.0724595
8000	0.64839834	1.6975222	0.65079004	1.0431184
10,000	0.7417457	1.7784712	0.74250895	1.0478663
Gradual Drift Anomaly (Network Size m)	$\mu$ (Anomaly)	$\sigma$ (Anomaly)	$\mu$ (Wavelet dB[12])	$\sigma$ (Wavelet dB[12])
2000	0.07693122	0.910007	0.076246604	0.4898912
4000	0.07699456	0.9098382	0.0763265	0.49061427
6000	0.07699457	0.908578	0.07632571	0.49055964
8000	0.07699094	0.9098535	0.07632209	0.49055964
10,000	0.07699085	0.9098535	0.07632202	0.49055642

## 5.2 Results and Discussion

This section shows the results of the analysis of anomaly detection approaches discussed in Section 4.1.2. The detection performances of the different approaches are obtained by training and testing them for a specific category of anomaly or in the presence of all the anomaly types.

In this study, three types of the anomaly are simulated with CAV dataset with varying anomaly duration  $d$ , scaling factor denoted as  $c_i$  for varying anomaly network densities distribution, and the anomaly rate denoted as  $\eta$ . The performances of CNN, BDL, and the proposed approach (DWT-BDL) are drawn from the variation of these parameters in detecting and identifying abnormal sensor behaviors. The detection approaches are implemented with ML libraries, namely TensorFlow and TFP, and Python packages. The dataset was trained, validated, and tested in 60 %, 20 %, and 20 % ratio, respectively. The validation and training sets are used to tune the parameters of the selected detection approaches, and different test sets are used to assess and measure the performance of the detection approaches. Each simulation is repeated 15 times with different seeds to achieve statically invariant outcomes for higher confidence. The performance of the selected approaches is evaluated in terms of accuracy, precision, sensitivity, and  $F1 - score$ .

### 5.2.1 Comparison of the Proposed DWT-BDL Approach and Conventional Approaches Under Single Anomaly System

This section evaluates the performances of the proposed DWT-BDL approach and the conventional detection approaches under a single anomaly system. Performance evaluations are carried out on the densities and duration of an anomaly in the CAV network. Different datasets, each with a specific type of anomaly, are generated, with anomaly incidence rate  $\eta$ , duration of anomaly  $d$ ,

and scaling factor  $c_i$  for varying network densities. The performance evaluations of the detection approaches are carried out on varying anomaly densities and duration on the three types of anomalies.

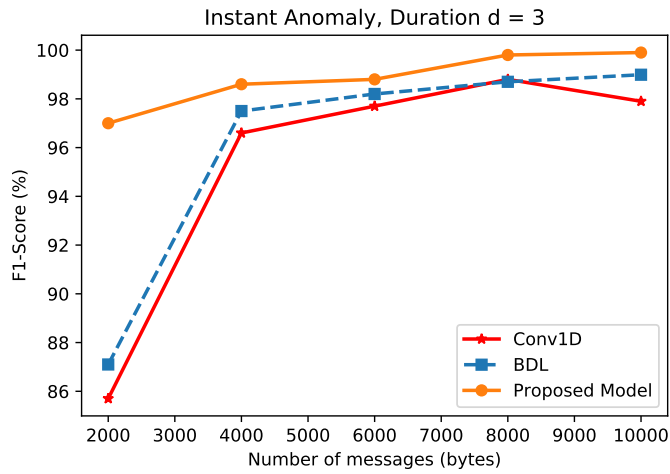
### Impact of Network Density on Anomaly Detection

#### (i) Instant anomaly

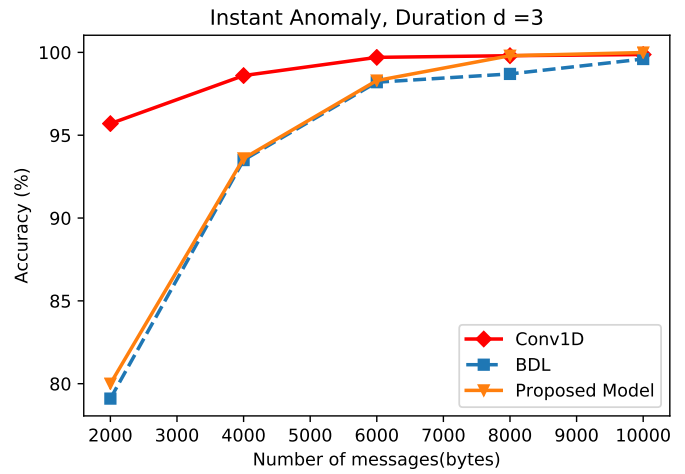
Figure 5.4 shows the performance of the anomaly detection approaches at different densities of anomaly distribution. At a low density of anomaly distribution, the proposed approach performs better than the SOTA approaches in all scenarios of instant anomaly distributions. For instance, for  $c \times \mathcal{N}(0, 0.01)$  with  $c = 2000$ , CNN and BDL have the performance values of 85.70 %, 85.00 %, 85.00 % and 87.10 %, as can be seen in Figures 5.4a and 5.4c. The same Figures demonstrate that at the same condition, the proposed approach shows improvement over SOTA approaches, with performance gains of 4.3 % and 2.7 % compared to CNN and 2.9 % and 1.9 % compared to BDL. Similarly, Figure 5.4d also illustrates the superiority of the proposed approach. However, the CNN approach in this scenario maintained a lead performance in some cases of  $c \times \mathcal{N}(0, 0.01)$  in the simulation, as shown in Figure 5.4b.

At high values of  $c \times \mathcal{N}(0, 0.01)$ , the proposed approach demonstrates superior performance over BDL and CNN in all the performance metrics except in few cases of the accuracy metric (see Figures 5.4b), where CNN indicates slight performance gains. However, the proposed approach outperforms CNN and BDL in all the cases of  $c \times \mathcal{N}(0, 0.01)$ . For instance, at  $c = 10000$ , the proposed approach's sensitivity shows performance gains of 5.00 % and 4.00 % compared to BDL and CNN, respectively. The proposed approach shows significant performance across all the performance metrics, as seen in Figure 5.4.

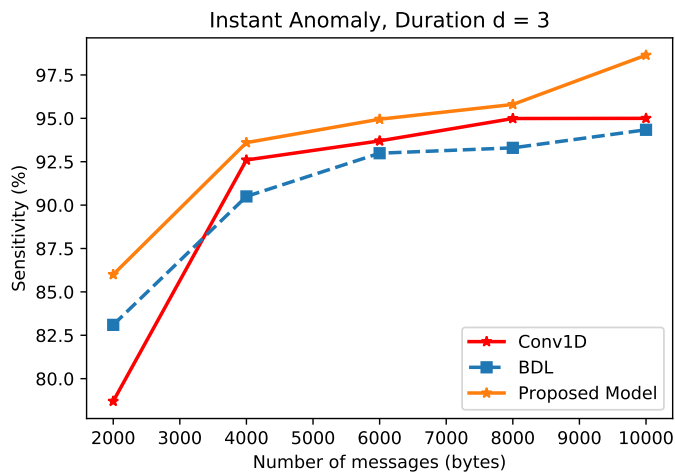
It can be generalized that as the density of anomaly distribution decreases, all the detection approaches exhibit low performance. This low performance in anomaly detection indicates the relevance of  $c$  in the anomaly detection system. The consistent superior performance of the proposed approach in all the metrics is due to the combination of BDL and DWT, which enhances the anomaly detection capability. The proposed approach utilizes the decomposition and denoising capability of DWT and couples with the robust BDL approach for decision-making.



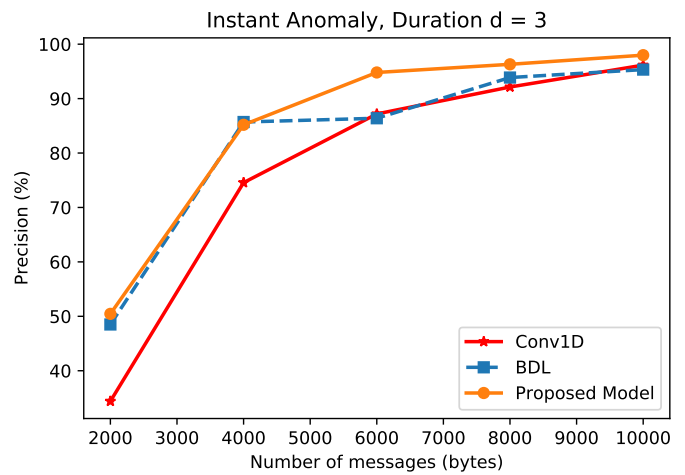
(A) Performance variation of  $F1 - score$  metric based on the number of messages



(B) Performance variation of accuracy metrics on the number of messages



(C) Performance variation of sensitivity metric based on the number of messages



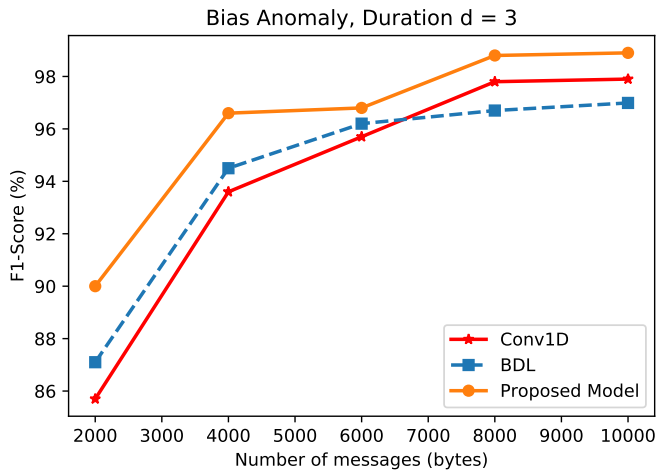
(D) Performance variation of precision metric based on the number of messages

FIGURE 5.4: Detection performance of CNN, BDL, and the proposed approach during the instance anomaly scenario

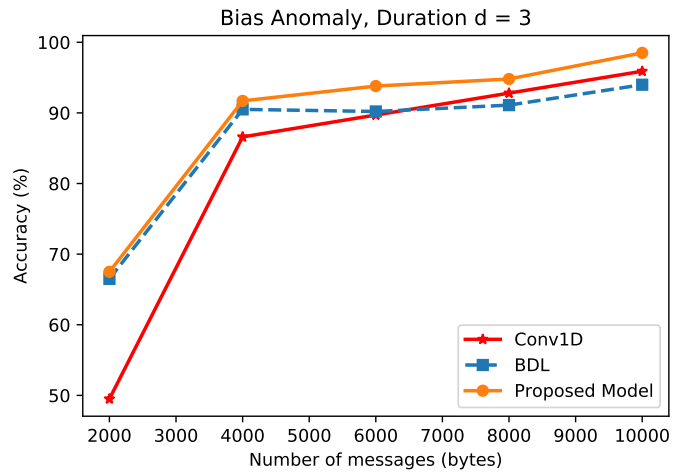
(ii) Bias anomaly

Figure 5.5 presents the performance results of BDL, CNN, and the proposed detection approach for bias anomaly. As demonstrated in the simulations, at low  $\mathcal{U}(0, c)$ , the BDL approach performs better than the CNN. Further, the proposed approach outperforms both BDL and CNN approaches in all the metrics at low anomaly density distributions. For example, as shown in Figure 5.5b, BDL improves in accuracy metric by approximately 5.2% in comparison to CNN, while the proposed approach displays performance gains of 1.2% and 5.2% over BDL and CNN, respectively, when  $c = 2000$  samples are drawn from  $\mathcal{U}(0, c)$ .

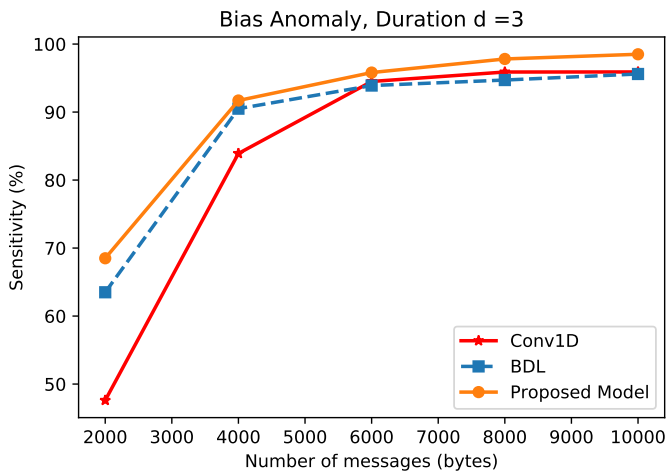
Moreover, at  $c = 10,000$  in the simulation, the efficiency of the detection approaches increases as the density of anomaly distribution increases in the CAV network. Detection approaches in this anomaly (i.e., attack) case show similar behavior in the instant anomaly case. The distribution is drawn from a fixed random variable  $\mathcal{U}(0, c)$  and duration  $d = 3$ . As illustrated in Figure 5.5c, the proposed approach shows improvement in sensitivity metric over BDL and CNN by about 4.50% and 2.60%, respectively. Similarly, Figures 5.5a and 5.5d depict the performance of the detection approaches on  $F1 - score$  and precision metrics. Results indicate the superiority of the proposed approach over the baseline approaches for different densities of the anomaly distribution in the bias anomaly type.



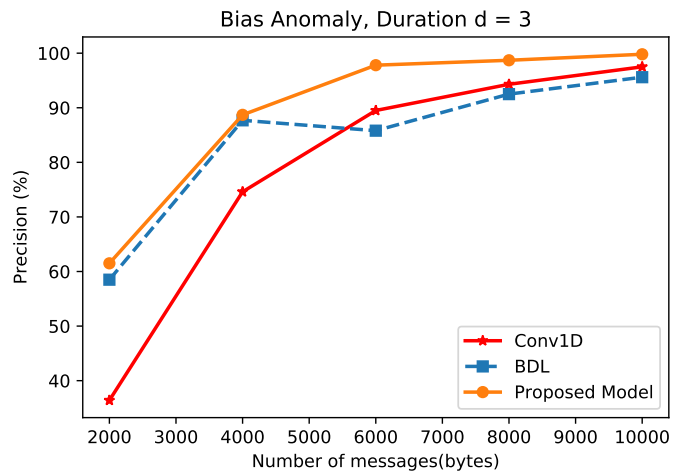
(A) Performance variation of  $F1 - score$  metric based on the number of messages



(B) Performance variation of accuracy metric based on the number of messages



(C) Performance variation of sensitivity metric based on the number of messages



(D) Performance variation of precision metric based on the number of messages

FIGURE 5.5: Detection performance of CNN, BDL, and the proposed approach during the bias anomaly scenario

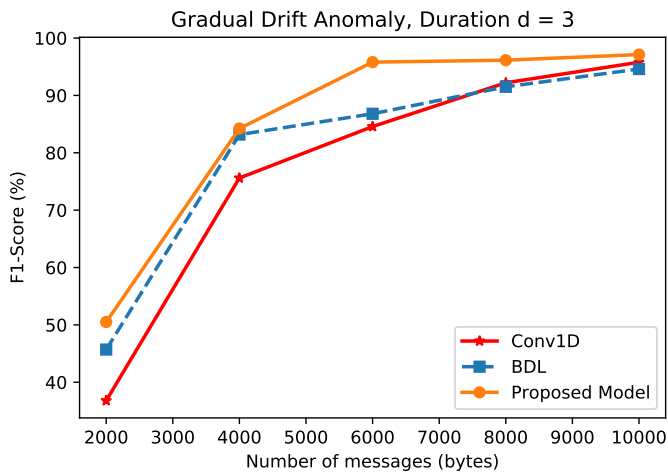
(iii) Gradual drift anomaly

Figure 5.8 shows the results of the BDL, CNN, and the proposed approach for gradual drift anomaly detection. This anomaly involves a linearly increasing set of values in the sensor reading, making it challenging to identify and discern the onset of abnormality from normal sensor values. We utilize a vector of linearly increasing values from 0 to  $c$ , which is denoted by  $\text{linspace}(0, c)$ . In general, for a low density of  $\text{linspace}(0, c)$  in the network, BDL outperforms the CNN. Figures 5.6a and 5.6d depict the BDL approach's performance at  $c$  value of 2000 for  $F1 - score$  and precision metrics, where the BDL approach outperforms the CNN approach by about 2.00 % and 14.10 %, respectively. However, at a high  $\text{linspace}(0, c)$  in the network, CNN consistently outperforms BDL in all the simulation outputs. For instance, when the  $\text{linspace}(0, c)$  is scaled by  $c = 10,000$ , CNN's  $F1 - score$  and precision metrics improve by 2.4 % and 3.00 %, respectively, over the BDL approach.

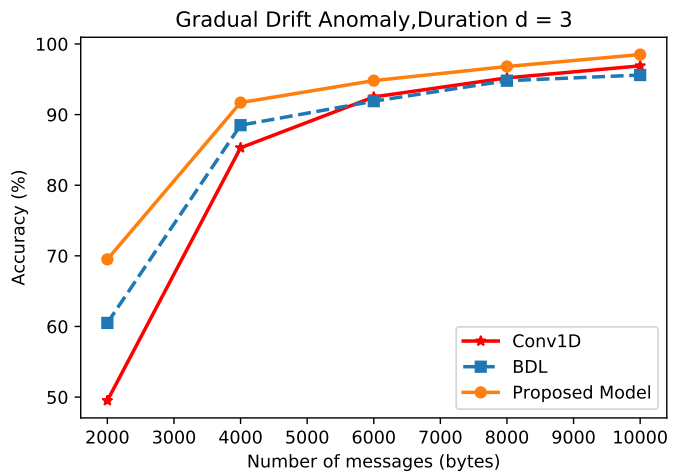
Moreover, the proposed approach's anomaly detection performance for gradual drift anomaly shows a significant improvement at low and high density of the anomaly distributions compared to the baseline approaches. For instance, as illustrated in Figures 5.6a, 5.6d, and 5.6c, the proposed approach shows performance gains of about 9.00 %, 16.00 %, and 20.00 % over CNN approach and about 6.95 %, 1.95 % and 2 % over BDL, in  $F1 - score$ , precision, and sensitivity metrics, respectively. Further, at a high value of  $c$ , the approach's detection performance increases across all the metrics. By intuition, larger anomaly distribution facilitates the extraction of meaningful information required by the detection approaches to detect the presence of an anomaly in the CAV network. In general, the proposed approach outperforms CNN and BDL. For instance, at  $c = 10,000$ , the proposed approach demonstrates superior performance in the  $F1 - score$ , precision, and sensitivity metrics by about 2.40 %, 1.80 %, and 2.60 % and



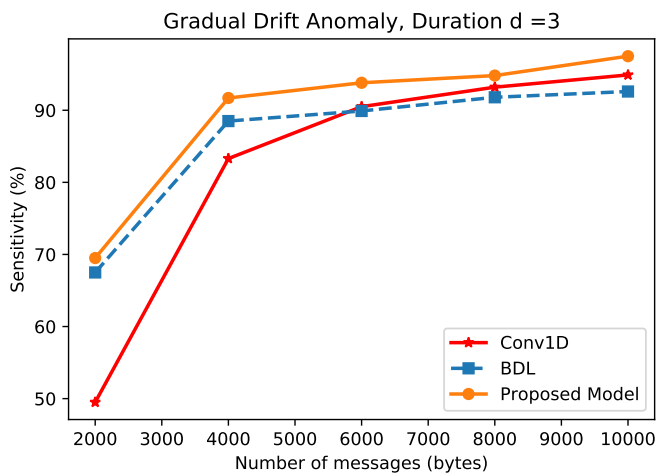
2.41 %, 2.68 %, and 4.90 % over CNN and BDL approaches, respectively. In addition, the proposed approach improves upon the detection performance of both CNN and BDL.



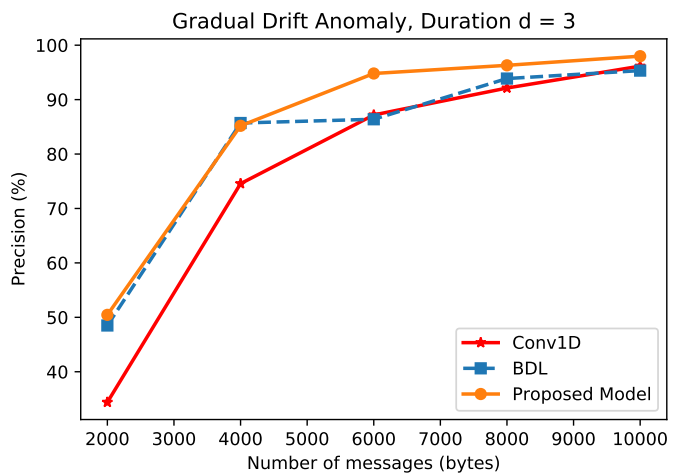
(A) Performance variation of  $F1 - score$  metric based on the number of messages



(B) Performance variation of accuracy metric based on the number of messages



(C) Performance variation of sensitivity metrics based on the number of messages



(D) Performance variation of precision metric based on the number of messages

FIGURE 5.6: Detection performance of the CNN, BDL, and the proposed approach during the gradual drift anomaly scenario

## Impact of Attack Duration

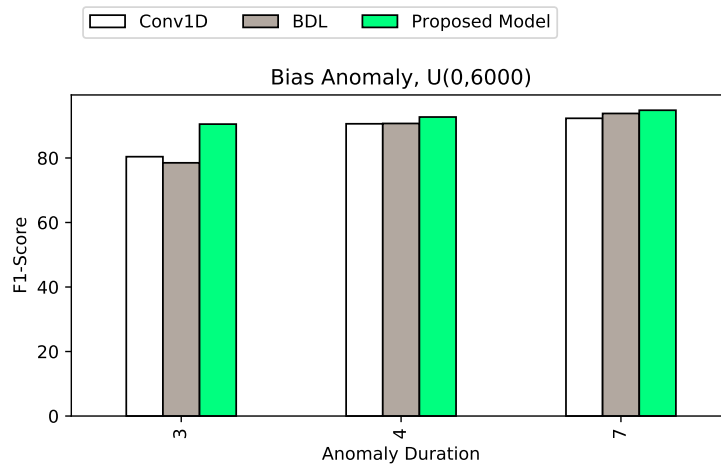
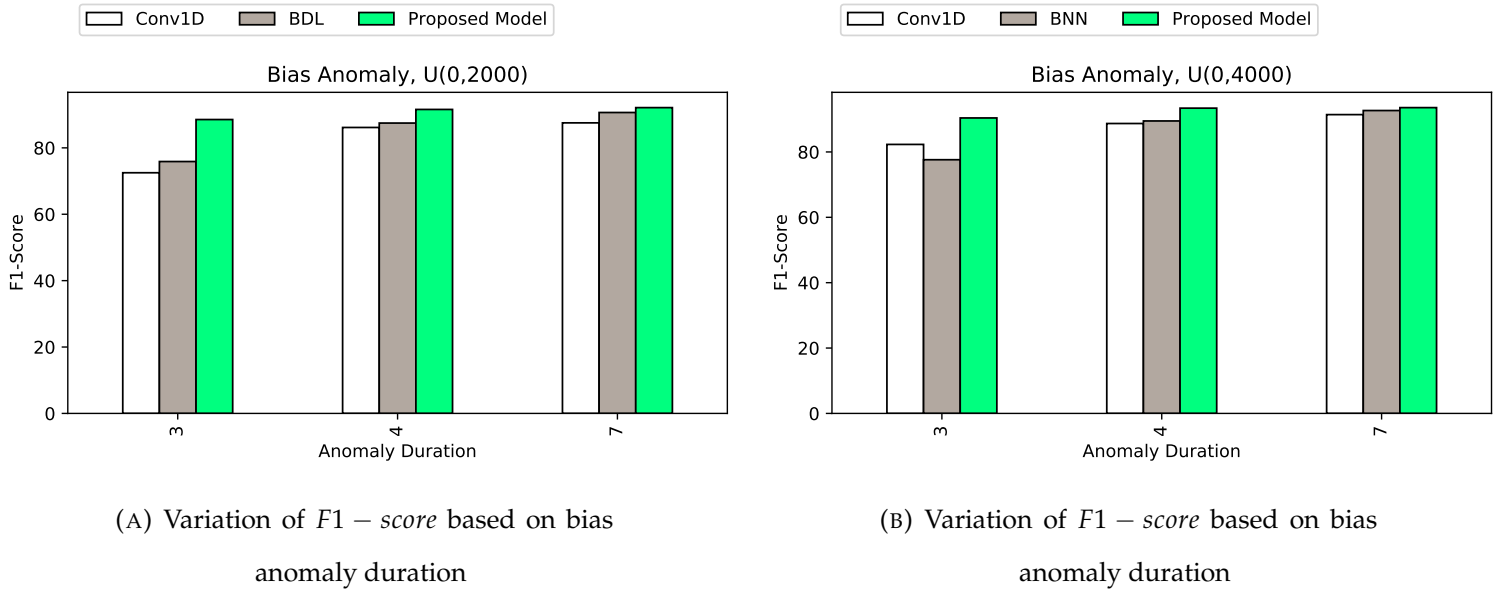
A set of simulations is carried out to demonstrate the performances of the different detection approaches with respect to varying anomaly duration  $d_i \in (3, 4, 7)$  at fixed density of anomaly distribution  $\mathcal{U}(0, c)$ , with  $c_i \in (2000, 4000, 6000)$  for the bias and gradual drift types of anomaly. As indicated in each of the subplots in Figure 5.7 – 5.9, for bias anomaly, each of the subplots has a varying anomaly duration at fixed anomaly density distribution. Also, Tables 5.2–5.4 illustrate the same experimental setting on gradual drift anomaly type with varying anomaly duration and fixed density of anomaly distribution.

## Bias Anomaly Setting

Figures 5.7a–5.7c depict the performances of the detection approaches in terms of the  $F1$  – score. Figure 5.7a shows that the detection performances of the different approaches considered in this context increase with a longer duration of anomaly, with anomaly distribution drawn from a fixed  $\mathcal{U}(0, c)$  distribution. Moreover, in Figures 5.7b and 5.7c, the proposed approach shows a similar pattern of improvement in detection performance with an increase in  $d$  and a different fixed  $\mathcal{U}(0, c)$ . In general, results from the selected performance metrics show that the different approaches' overall detection efficiency increases consistently with increase in the magnitude of both  $d$  and  $\mathcal{U}(0, c)$  distribution. For instance, Figure 5.7c with anomaly distribution  $\mathcal{U}(0, 6000)$  shows better performance than Figures 5.7a and 5.7b with distributions  $\mathcal{U}(0, 2000)$  and  $\mathcal{U}(0, 4000)$ , respectively, with the same  $d_i \in (3, 4, 7)$ .

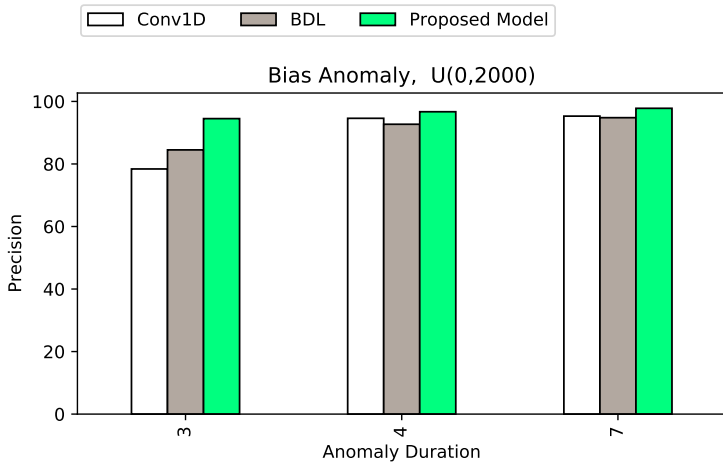
Further, in Figures 5.8a–5.8c and Figures 5.9a–5.9c, the detection approaches' sensitivity and precision results consistently improve with the increase in both in  $d$  and  $\mathcal{U}(0, c)$ . However, an exception occurs in Figures 5.8b and 5.8c with anomaly duration  $d = 7$ . In this case, an increase in the anomaly duration and distribution appears not to affect the proposed approach's

precision results since the detection approach could not detect the variation in the two anomaly distributions. However, the misclassification rate presented in both cases is not significant.

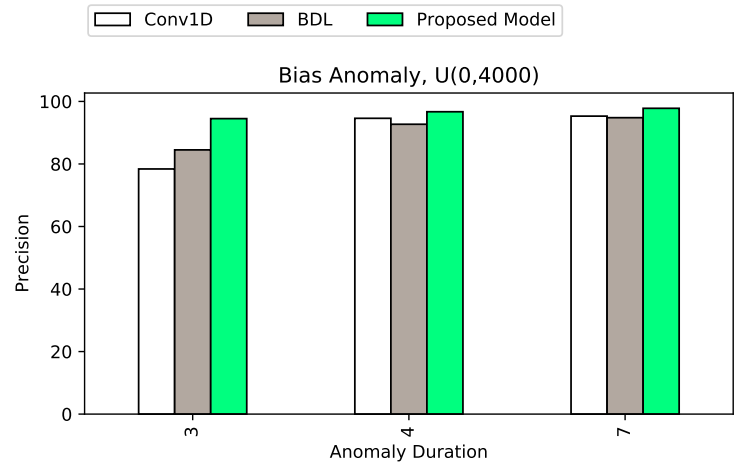


(C) Variation of  $F1 - score$  based on bias anomaly duration

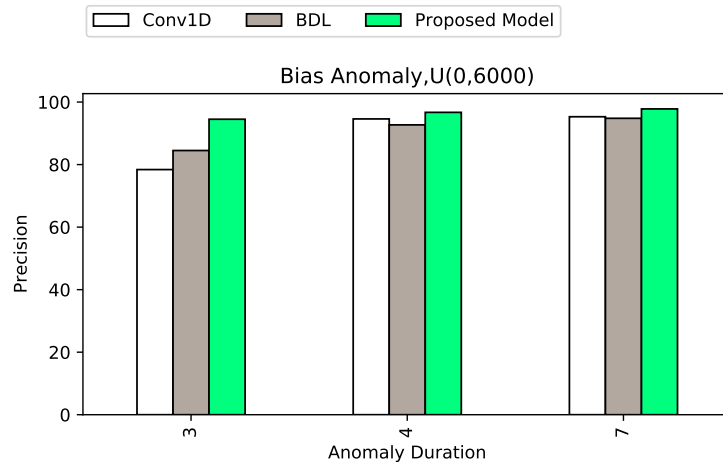
FIGURE 5.7: Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions during the bias anomaly scenario



(A) Variation of precision based on bias anomaly duration

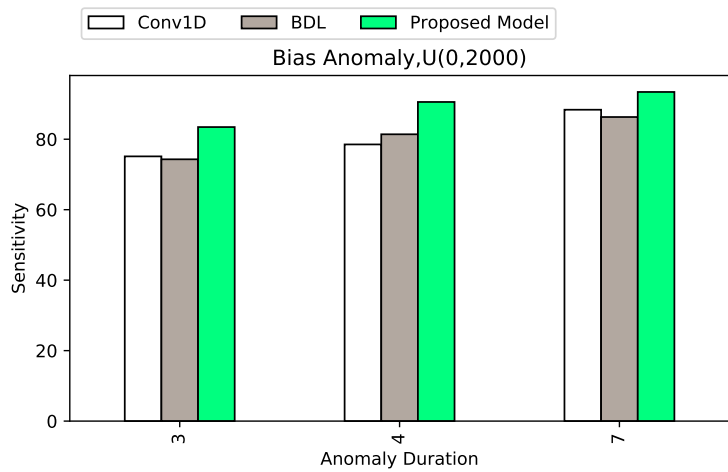


(B) Variation of precision based on bias anomaly duration

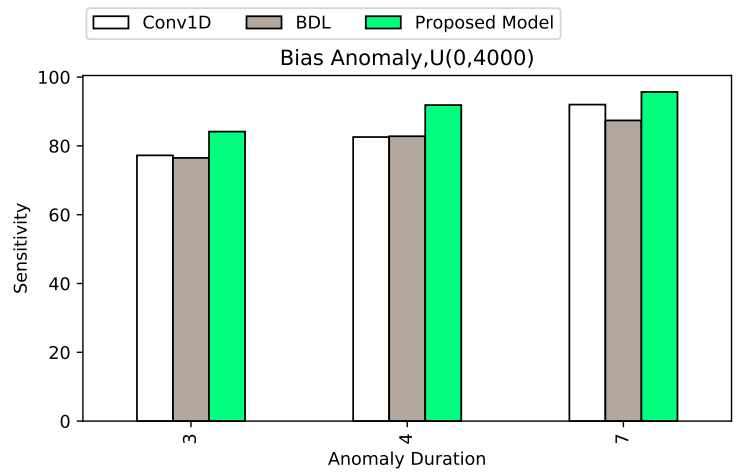


(C) Variation of precision based on bias anomaly duration

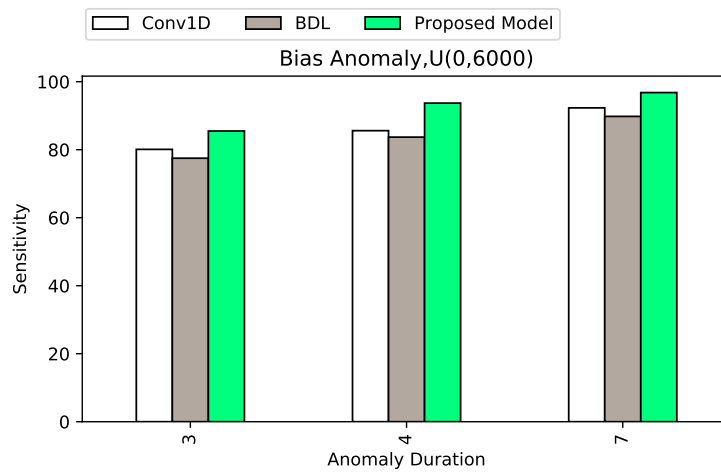
FIGURE 5.8: Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions during the bias anomaly scenario



(A) Variation of sensitivity based on bias anomaly duration



(B) Variation of sensitivity based on bias anomaly duration



(C) Variation of sensitivity based on bias anomaly duration

FIGURE 5.9: Detection performance of the BDL, CNN, and proposed approach for different anomaly duration/distributions in bias anomaly scenario

## Gradual Drift Anomaly Setting

A similar simulation is carried out for gradual drift anomaly. The simulation uses a vector of linearly increasing values from 0 to  $c_i \in (2000, 4000, 6000)$ , denoted by the  $\text{linspace}(0, c)$  function. Also, the various values of  $d$ ,  $d = \{3, 4, 7\}$ , and the density of the anomaly distribution,  $\text{linspace}(0, c)$ , are taken into account.

Tables 5.2–5.4 show the results of the different approaches during the gradual drift anomaly scenario. The detection approaches generally do not display significant improvement in the detection capability compared with the bias anomaly cases. However, at high values of  $d$  and  $\text{linspace}(0, c)$ , the detection approaches appear to have an increase in detection strength. For instance in Tables 5.2–5.4, at low duration and magnitude of anomalous sensor distribution, BDL performs better than the CNN approach. However, the detection performance of the CNN approach increases with anomaly duration  $d$  and magnitude  $\text{linspace}(0, c)$ . The effect of the sliding window function in the CNN approach can increase the CNN performance.

As seen in Tables 5.2–5.4, the proposed approach outperforms both CNN and BDL approaches across all the experiments. For example, in row 3 of Table 5.2, the proposed approach's sensitivity, precision, and  $F1$  – score metrics increase by 6.98%, 7.37%, and 7.32%, respectively, compared to the CNN approach and by 9.10%, 11.89%, and 9.37%, respectively, compared to the BDL approach. Results in Table 5.3 also indicate that the proposed approach outperforms CNN and BDL. For instance, in the same row 3, the proposed approach's sensitivity, precision, and  $F1$  – score increase by 0.33%, 2.85%, and 4.8%, respectively, compared to the CNN and 3.18%, 5.68%, and 9.7%, respectively, compared to BDL.

Similarly, Table 5.4 presents the values of the detection approaches applied in the experiment with  $\text{linspace}(0, 2000)$ , where row 3 of the investigation

demonstrates the proposed approach's superior performance compared to BDL, and CNN approaches. For example, with the proposed approach, the sensitivity, precision, and *F1 – score* increase by 8.01%, 4.6%, and 9.33%, respectively, relative to CNN and by 3.96%, 3.74%, and 3.9%, respectively, relative to BDL.

		CNN (%)			BDL (%)			Proposed approach (%)		
	Duration, $d$	Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score
Anomaly distribution density	7	92.15	95.46	93.27	90.35	91.34	91.68	94.34	96.98	95.32
Base value + <i>inspace</i> (0, 6000)	4	90.10	91.86	88.54	87.57	88.60	88.63	93.17	95.7	93.54
Base value + <i>inspace</i> (0, 6000)	3	79.98	88.43	86.68	84.89	81.50	84.63	91.87	90.67	94.00

TABLE 5.2: Detection performance of the approaches during the drift anomaly scenario, with  $\mathcal{U}$  (0, 6000)



Anomaly Distribution Density	Duration, $d$	CNN(%)			BDL (%)			Proposed approach (%)		
		Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score
Base value + $linspace(0, 4000)$	7	91.00	93.46	90.33	89.50	88.34	90.16	92.15	96.00	93.68
Base value + $linspace(0, 4000)$	4	82.57	91.86	86.35	87.41	87.70	87.52	89.20	94.7	90.42
Base value + $linspace(0, 4000)$	3	87.24	87.43	83.30	84.39	84.60	78.40	87.57	90.28	88.10

TABLE 5.3: Detection performance of the approaches during the drift anomaly scenario, with  $\mathcal{U}(0, 4000)$

		CNN (%)			BDL (%)			Proposed approach (%)		
Anomaly distribution density	Duration, $d$	Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score	Sensitivity	Precision	F1 - score
Base value + <i>linspace</i> (0,2000)	7	88.63	98.46	98.30	87.35	96.34	91.68	90.34	93.87	99.46
Base value + <i>linspace</i> (0,2000)	4	78.52	96.83	97.54	83.40	95.60	98.63	90.10	99.7	98.57
Base value + <i>linspace</i> (0,2000)	3	75.13	93.10	98.57	83.45	93.67	96.00	87.41	97.70	89.94

TABLE 5.4: Detection performance of the approaches during the drift anomaly scenario, with  $\mathcal{U}$  (0, 2000)

### **Discussion on the Performances of the Detection Approaches During the Single Anomaly Scenario**

As seen in the experiments, the performance of the detection approaches increases significantly with the increase in the value of anomaly duration  $d$  and considered network density distribution,  $c \times \mathcal{N}(0, 0.01)$ ,  $\mathcal{U}(0, c)$ , and  $\text{linspace}(0, c)$ . By intuition, a larger distribution causes a larger deviation from the true values of the normal sensors' behaviors. Thus, the greater the effectiveness of the approaches in extracting enough features to detect anomalies in the CAV network.

Moreover, the higher performance of the detection approaches with a longer duration can be attributed to the fact that a longer duration gives detection approaches the time to accumulate knowledge about the behaviors and anomaly impacts in a CAV network.

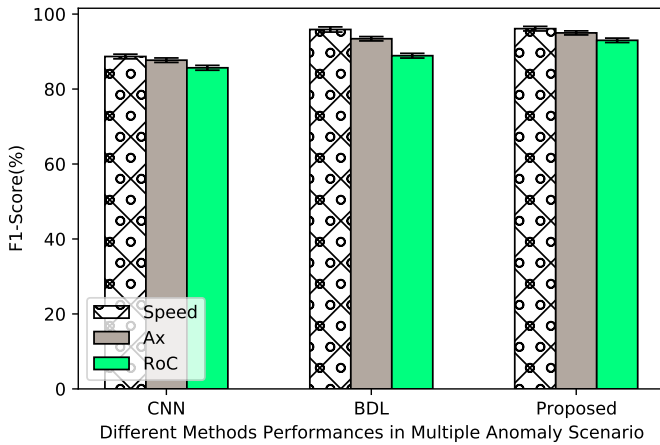
Overall, in the single attack system, as shown in Section 5.2.1, the detection approaches can generalize and correctly classify previous unseen observations with similar distribution (test set) throughout the experiment by training on representative training sets. However, CAV anomaly detection approaches can experience abnormalities for which they have not been specifically trained in practice. Details of the incidents of unseen observations are expressed in the multiple anomaly/attack scenarios discussed in Section 5.2.2.

## 5.2.2 Performance of the Approaches During the Multiple Anomaly Scenario

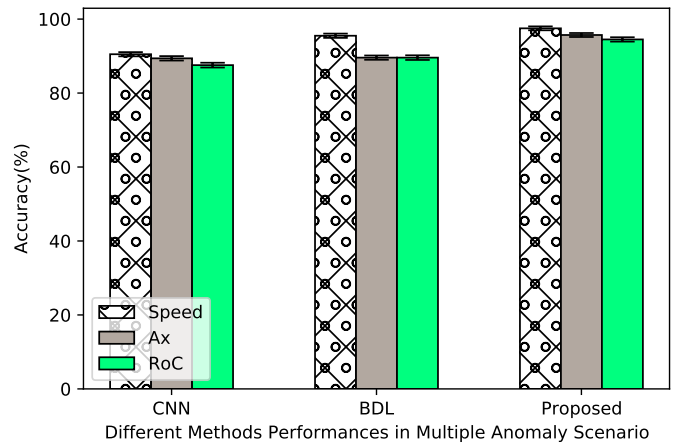
This section addresses a specific detection approach's effectiveness and reliability by subjecting it to multiple anomaly scenarios. Exposure to varying attributes of abnormal behaviors gives insight into the performances of these approaches. The attribute is achieved in the test dataset by integrating the three anomaly types discussed in Section 3.1.3. The multiple anomalies are modeled with  $100000 \times \mathcal{N}(0,7)$ ,  $\mathcal{U}(0,6000)$ ,  $linspace(0,6000)$ , and  $d = 7$ . Having been trained on one of the anomaly types, an investigation is carried out on generalizing the detection approaches during the unobserved multiple anomaly scenarios. In the experimental settings, the impacts of the  $\eta$  on the three different sensors are only considered. Evaluation performance of the detection approaches are carried out at  $\eta = 10\%$  and  $\eta = 50\%$ . Further, the simulation of the detection approaches are conducted several times to ensure statistical relevance by providing the mean performance of 95 % credible interval for BDL, DWT-BDL, and confidence intervals for CNN, respectively.

Figure 5.10 depicts the performance of the study's detection approaches at  $\eta = 50\%$  during the multiple anomaly scenario. The  $\eta$ -values are set high to capture the selected approach's behaviors and detection capabilities for the threat that can pose considerable risk to the CAV network's operation. Results indicate the performance variations of the detection approaches across the sensor readings in all the metrics in the simulations. In particular, for RoC sensor, it is observed that the approaches' performance values are worse compared with the Ax and speed sensors, which appear to show much smoother reading over-time under the same anomaly scenario. The RoC sensor behaviors are partly attributed to the tremendous variation in the consecutive readings. The results in

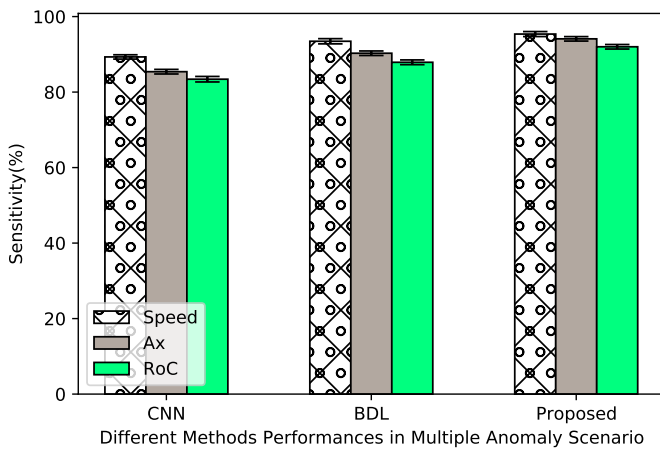
Figure 5.10 validate the performance degradation of the detection on the unobserved abnormal RoC sensor values. However, the proposed approach demonstrates a lead performance in all the scenarios with the values of  $4.92 \pm 0.009$  and  $6.95 \pm 0.009$  over BDL and CNN, respectively, considering the worst-case scenario of RoC sensor analysis. In the same vein, Figures 5.10b to 5.10d replicate the same performance improvement with the use of the proposed approach.



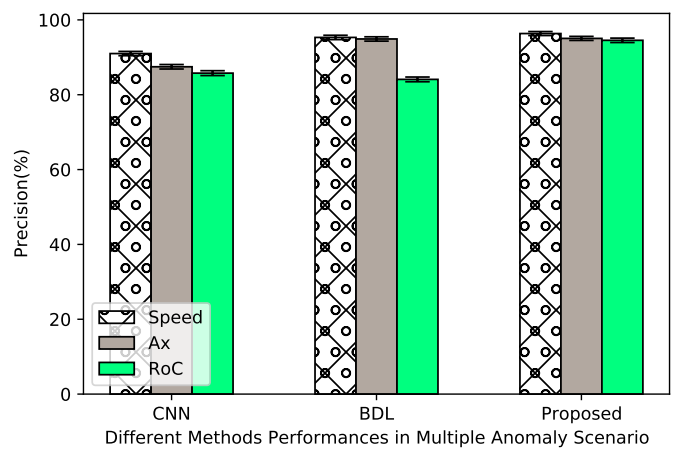
(A) Performance variation of the different approaches on the three sensors (speed, Ax, and RoC) in terms of *F1 – Score* metric



(B) Performance variation of the different approaches on the three sensors (speed, Ax, and RoC) in terms of accuracy metric



(C) Performance variation of the different approaches on the three sensors (speed, Ax, and RoC) in terms of sensitivity metric



(D) Performance variation of the different approaches on the three sensors (speed, Ax, and RoC) in terms of precision metric

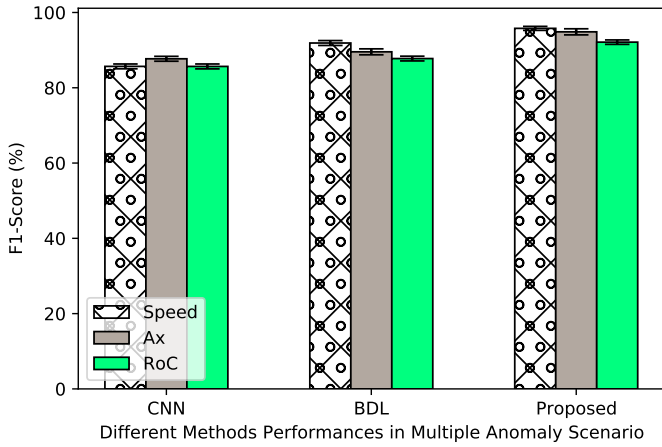
FIGURE 5.10: Detection performance at 95% CI and CRI across 15 to 20 different executions for all three approaches, at anomaly rate  $\eta = 50\%$  and in the presence of all the types of anomalies

A similar experiment is also conducted with all the anomaly types to investigate the behaviors of the detection approaches in the CAV system for a

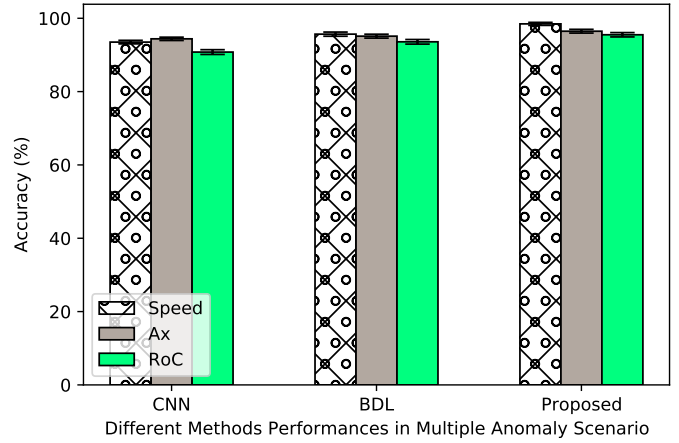
low value of  $\eta = 10\%$  and  $d = 7$ . As shown in Figure 5.11, the three selected approaches' detection performances vary among various sensors. Again, the three detection approaches show the lowest performance values when applied to abnormal RoC sensor (BSM) values at this simulation analysis stage. This could be attributed to the poor performance of the detection approaches to the variation in the input read sequence of RoC sensor (BSM) readings.

At  $\eta$  set to 10%, the approaches show a better performance classification accuracy. Generally, the approaches do not show significant improvement in performances on other metrics, especially in Figure 5.11a, when compared to  $\eta = 50\%$  in Figure 5.10a. The observation complies with intuition, as the lower value of  $\eta$  makes the anomaly more elusive and thus more challenging to detect. Conversely, a high detection accuracy, as shown in Figure 5.11b, may be due to the imbalanced nature of the BSM samples. Thus, the classification accuracy appears to favor the more representative class [80]. Accuracy metric may not be an appropriate performance metric for imbalanced data. Furthermore, as indicated in Figures 5.11c and 5.11d, the proposed approach demonstrates significant performance improvement in precision and sensitivity plots over BDL and CNN.

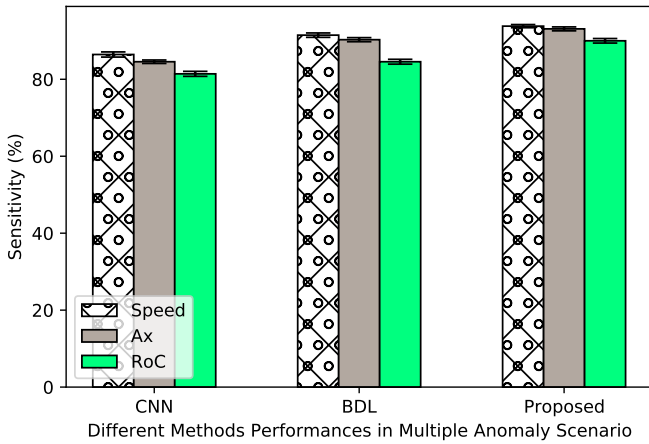
The performance evaluations of the detection approaches in this context of simulation are mainly focused on *F1 – score*, which provides more insights into the strength of the detection approaches in an imbalance class scenario [81]. From the results obtained for *F1 – score*, as presented in Figure 5.11a, it can be concluded that the BDL outperforms the CNN. Moreover, the proposed approach significantly improves performance over the BDL and CNN approaches, with values of  $4.36 \pm 0.010$  and  $6.45 \pm 0.010$ .



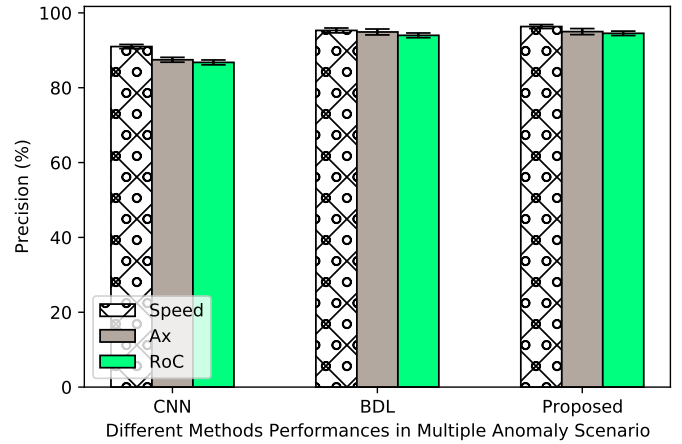
(A) Performance variation of the various approaches on the three sensors (speed, Ax, and RoC) in terms of *F1 – Score* metric



(B) Performance variation of the various approaches on the three sensors (speed, Ax, and RoC) in terms of accuracy metric



(C) Performance variation of the various approaches on the three sensors (speed, Ax, and RoC) in terms of sensitivity metric



(D) Performance variation of the various approaches on the three sensors (speed, Ax, and RoC) in terms of precision metric

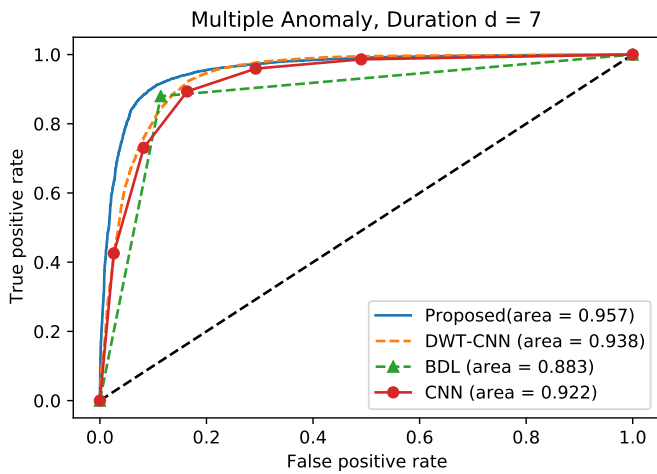
FIGURE 5.11: Detection performance at 95% CI and CRI across 15 to 20 different executions for all three approaches, at anomaly rate  $\eta = 10\%$  and in the presence of all the types of anomalies

Furthermore, in the simulation setting, CNN is subjected to the same DWT denoising, and the performance is validated with the proposed approach. The

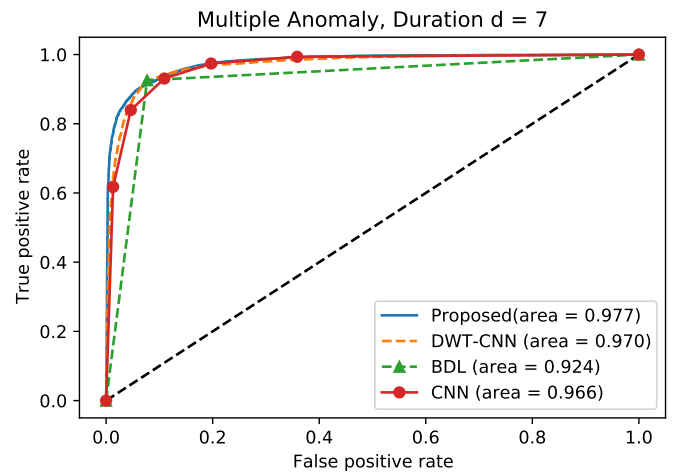


AUC of the receiver operating characteristics curve is also computed to validate the proposed approach's performance and reliability during gradual drift anomaly scenario at  $\eta = 10\%$  and  $\eta = 50\%$ , respectively.

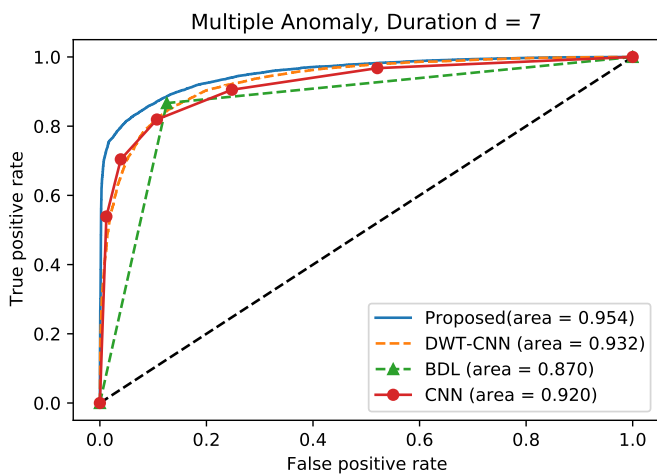
In the simulation setting, the detection approach is trained based on the bias and instant anomaly types, and the trained detection approaches are applied on the gradual drift anomaly dataset. Then, the performance of the trained approaches on the gradual drift anomaly test set is validated. Figures ?? and 5.12 show that the proposed approach demonstrates superior performances than BDL, CNN, and DWT-CNN. For instance, in Figure 5.13a, at  $\eta = 10\%$ , the proposed approach displays a performance gain of 1.9%, 7.4%, and 3.5% over DWT-CNN, BDL, and CNN, respectively. At the same time, Figure 5.13b shows that when  $\eta = 50\%$ , the proposed approach displays improved values of 2.2%, 8.4%, and 3.4% compared with DWT-CNN, BDL, and CNN approaches, respectively. Similarly, Figure 5.13c indicates that the proposed approach demonstrates a significant improvement through values of 0.7%, 5.3%, and 1.1% compared with DWT-CNN, BDL, and CNN approaches, respectively. Finally, Figure 5.13d also proves that the proposed approach provides a significant improvement in performance with values of 2.1%, 0.4%, and 2.6% over DWT-CNN, BDL, and CNN, respectively.



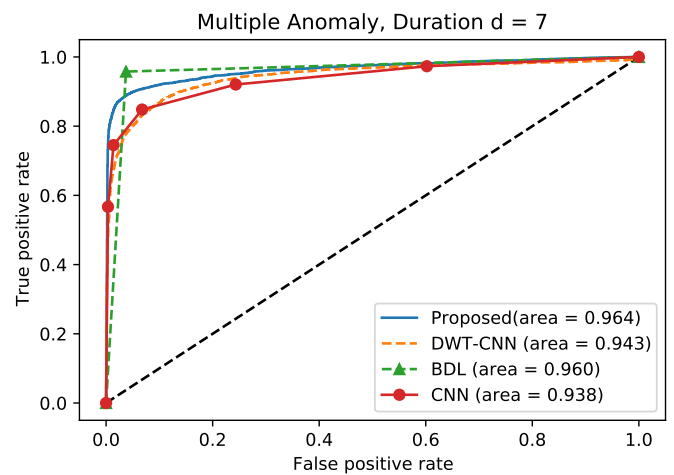
(A) Performance variation of the various approaches in terms of RoC metric during the gradual drift anomaly, at  $\eta=10\%$ .



(B) Performance variation of the various approaches in terms of RoC metric during the gradual drift anomaly, at  $\eta = 50\%$ .



(C) Performance variation of the various approaches in terms of RoC metric during the gradual drift anomaly, at  $\eta=10\%$ .



(D) Performance variation of the various approaches in terms of RoC metric during the gradual drift anomaly, at  $\eta=10\%$ .

FIGURE 5.12: Performance variation of the various approaches trained on bias anomaly during the gradual drift anomaly, at  $\eta = 10\%$  and  $\eta = 50\%$

## **Discussion on the Performances of the Detection Approaches During Multiple Anomaly Scenario**

The experiment aims to establish a likely situation of multiple anomalies (i.e., attacks) that depicts real-life scenarios vulnerable to multiple anomalies/attacks. The proposed approach demonstrates good performances due to the prior probability of the Bayesian approach, which establishes synergies/fusions among heterogeneous information, which in turn aid in the classification of out-of-distribution instances as unknowns. The synergy/fusion strategy provides impressive detection capability to detect unknown anomalies/attacks in a CAV network [82], [83].

### **5.2.3 Approaches Under Single Anomaly System Under DWT-BDL Proposed Approach**

Further evaluation is made with the comparison of the proposed DWT-BDL with the combined convolutional neural network with attention-based long short term memory (CNN-ALSTM) [84] and Kalman filter-convolutional neural network (KF-CNN) [84].

The performance of the three approaches are measured for F1-score, precision and sensitivity. F1-score is the harmonic mean of sensitivity and precision metrics, respectively. In comparison, precision is the proportion of abnormal sensor values among those predicted to be anomalous. Sensitivity access is the proportion of the correctly detected abnormal sensor values from the total number.

The proposed DWT-BDL approach during the comparative analysis with the CNN-ALSTM and KF-CNN approaches are evaluated on the test sample of the Gradual drift anomaly dataset, and the results are shown in Tables 5.5 and 5.6 respectively. Gradual drift anomaly type is selected to know the strength of

the proposed approach with baseline approaches. Each approach is executed on the abnormal sensor reading (Gradual drift) ten times, and the mean value and the confidence interval (C.I.) are computed. The confidence interval can be computed as follows:

$$\text{Confidence interval(C.I.)} = \bar{x} \pm z * \frac{\sigma}{n} \quad (5.1)$$

where  $\bar{x}$  is the mean values of the different output results of the metrics,  $z$  is the  $z$ -score,  $\sigma$  is the standard deviation and  $n$  is the sample size of the data. The  $\sigma$  is obtained as follows:  $\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$  where  $N$  is the number of different simulation outputs considered in the simulation,  $x_i$  is the output of the individual simulation output and  $\bar{x}$  is the mean based on the number of simulations.

In the simulation, emphasis is laid on the approach's performance on the F1-score, and sensitivity metrics for classification and the performance values of the metrics are highlighted in bold font in the two cases of gradual anomaly distributions ( $\text{linspace}(0, 2000)$ ,  $\text{linspace}(0, 10000)$ ). F1-score provides more insight on the strength of a detection mechanism in the event of imbalance dataset [81], while sensitivity directly impacts the reliability of fused data in CAV [22]. At a very high anomaly distribution, the detection approach's performances increase across all the metrics. By intuition, larger anomaly distribution facilitates the extraction of meaningful information required to detect the presence of an anomaly in the CAV network [85], [22] At  $\text{linspace}(0, 10000)$ , the proposed approach demonstrates superior performance in the **F1 – score** and sensitivity metrics as shown with the bold font in Table 5.5 with values of  $97 \pm 0.008$ , and  $97.08 \pm 0.009$  for F1-score and sensitivity metrics. Though in this case of anomaly distribution, CNN-ALSTM has a better precision performance than the DWT-BDL and KF-CNN approaches. However, the proposed DWT-BDL

approach has a more precise value of (C.I.), which indicates a more reliable output. A similar simulation is carried out at a low density of gradual drift anomaly distribution of  $linspace(0,2000)$ . As demonstrated in Table 5.6 with bold font, the proposed approach performs better than the state-of-the-art (SOTA) with  $89.39 \pm 0.035$  and  $87.13 \pm 0.016$  in F1-score and sensitivity metrics, respectively.

TABLE 5.5: Detection Performance and the 95 % confidence interval across 10 different executions for the three approaches, at anomaly rate of 10 %, duration 7 with respect to Gradual drift anomaly type with anomaly distribution of  $linspace(0,10000)$ .

Approach	Metric	Mean ( $\bar{x}$ )	C.I.
DWT-BDL	F1-score	<b>97.60</b>	<b>97.60 <math>\pm</math> 0.008</b>
	Precision	96.30	96.30 $\pm$ 0.006
	Sensitivity	<b>97.08</b>	<b>97.08 <math>\pm</math> 0.009</b>
CNN-ALSTM	F1-score	93.45	93.45 $\pm$ 0.017
	Precision	96.67	96.67 $\pm$ 0.009
	Sensitivity	94.51	94.51 $\pm$ 0.015
KF-CNN-1D	F1-score	93.39	93.39 $\pm$ 0.019
	Precision	96.15	96.15 $\pm$ 0.016
	Sensitivity	95.64	95.64 $\pm$ 0.014

TABLE 5.6: Detection Performance and the 95 % confidence interval across 10 different executions for the three approaches, at anomaly rate of 10 %, duration of 7 with respect to Gradual drift anomaly type with anomaly distribution of  $linspace(0,2000)$ .

Approach	Metric	Mean ( $\bar{x}$ )	C.I.
DWT-BDL	F1-score	<b>89.38</b>	<b>89.38 ± 0.035</b>
	Precision	93.29	93.29 ± 0.023
	Sensitivity	<b>87.13</b>	87.13± 0.016
CNN-ALSTM	F1-score	87.28	87.28 ± 0.037
	Precision	92.92	92.92 ± 0.021
	Sensitivity	82.65	<b>82.65±0.021</b>
KF-CNN-1D	F1-score	85.48	85.48 ± 0.042
	Precision	88.28	88.28 ± 0.056
	Sensitivity	81.32	81.32±0.002

In addition, the complexity of detection approach is attributed to the training time as indicated in [86]–[88]. This assumption is made in this context. From the training time for the two gradual drift anomaly distributions as indicated in Table 5.7, our proposed DWT-BDL takes a longer time in its training process. At the same time, CNN-ALSTM has a shorter training process with values of **147**, and **149** as indicate with bold font in the two gradual drift anomaly cases. It is assumed that the proposed approach is more complex than the baseline approach in this context.

TABLE 5.7: Approach complexity based on training time

Gradual drift anomaly distribution	Approach	Training Time(s)
<i>linspace(0,2000)</i>	DWT-BDL	282
	CNN-ALSTM	<b>147</b>
	KF-CNN	153
<i>linspace(0,10000)</i>	DWT-BDL	297
	CNN-ALSTM	<b>149</b>
	KF-CNN	156

#### 5.2.4 Approaches Under Multiple Anomaly System Under DWT-DDQN Proposed Approach

This section first runs simulations to select the best discount factor,  $\gamma$  value, that presents the best reward with minimum loss. A series of simulations of various  $\gamma$  values ranging from 0 to 1 are conducted. The finding shows that  $\gamma$  values that fall within the range, as seen in Figure 5.13, provide the desired outcomes, with  $\gamma = 0$  being the best. If  $\gamma = 0$ , the agent would be completely naive and only learn about actions that produce an immediate reward. If  $\gamma = 1$ , the agent would weigh each of its actions against the total of all future rewards [89]. Here, the performance metrics are *F1 – score* and sensitivity. The value of  $\gamma = 0$  is kept constant during the modeling, which fits the unstable topology of the CAVs network, where most actions do not have long-lasting consequences.

The gradual drift anomaly type is also adopted as discussed in Section 5.2, with the same parameters, to test the capability of the approach proposed in Section 4.1.4 and compare its performances with the available approaches discussed in Sections 4.1.5–4.1.6. Here, EMLP and SVM approaches are selected because of their extensive use in CAV anomaly detection [90].

In Figures 5.14 and 5.15, the results in terms of  $F1 - score$  and sensitivity metrics show that the proposed DWT-DDQN approach performs better than EMLP and SVM approaches. This is because the proposed DWT-DDQN approach exploits both DWT's and DDQN's ability to enhance results. Figure 5.14a indicates that at  $\eta = 10\%$ , the abnormal readings become more difficult to detect. However, at  $c = 10000$ , the DWT-DDQN provides a significant improvement (up to 10% and 20%) in  $F1 - score$  when compared to SVM and EMLP. The proposed approach achieves a promising performance fit due to the combination of DWT and DDQN features. EMLP and SVM approaches could not have good results at this density of anomaly distribution simply because EMLP, for example, does not show strong knowledge of dependence among sequences, which is important in time series datasets like BSMs [91]. Moreover, SVM could not detect anomalies at a small value of  $\eta$  since it depends on label; in other words, SVM lacks the quality of extracting meaningful information from data features to enable it to have good detection capability at small anomaly rates [11].

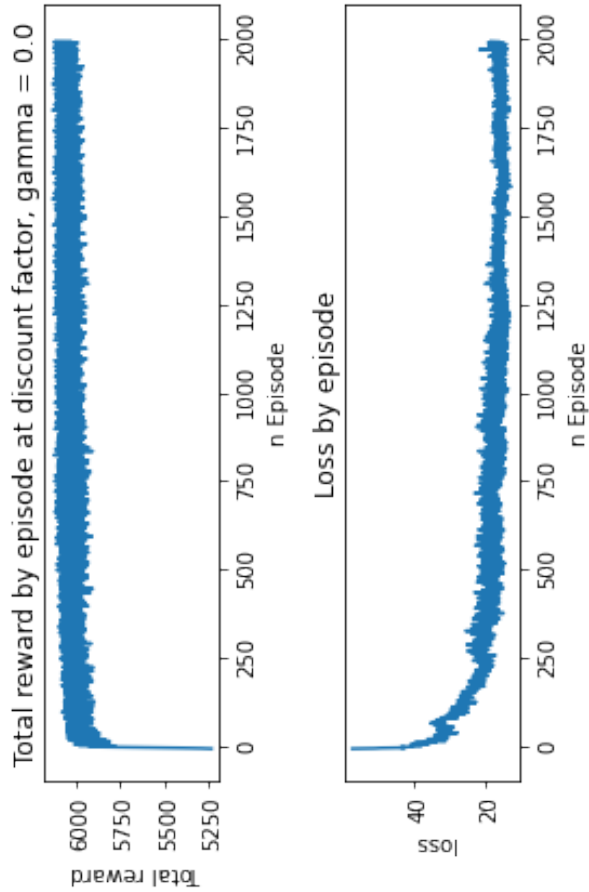
The performance of the proposed approach to an  $\eta$  value ( $\eta = 50\%$ ) that may pose a significant risk to the operation of the CAV network is further validated. At this point, results show a better performance across approaches in  $F1 - score$  metric when compared with  $\eta = 10\%$ . As seen in Figure 5.14a, MLP and SVM approaches show similar performances at  $c = 10,000$ . However, the approaches show better performance in terms of  $F1 - score$  when compared with a performance at  $\eta = 10\%$ . This improvement in detection could be due to the increase in the anomaly involved in the CAV network at this point. Results prove that the proposed approach outperforms EMLP and SVM in this case, with efficiency gains of 7% at  $c = 10,000$  for  $F1 - score$  metric.

Similarly, Figures 5.15a and 5.15b present performances in terms of sensitivity metric. It can be seen that DWT-DDQN demonstrates superior outcomes than EMLP and SVM at both  $\eta = 10\%$  and  $\eta = 50\%$ . In Figure 5.15a, the

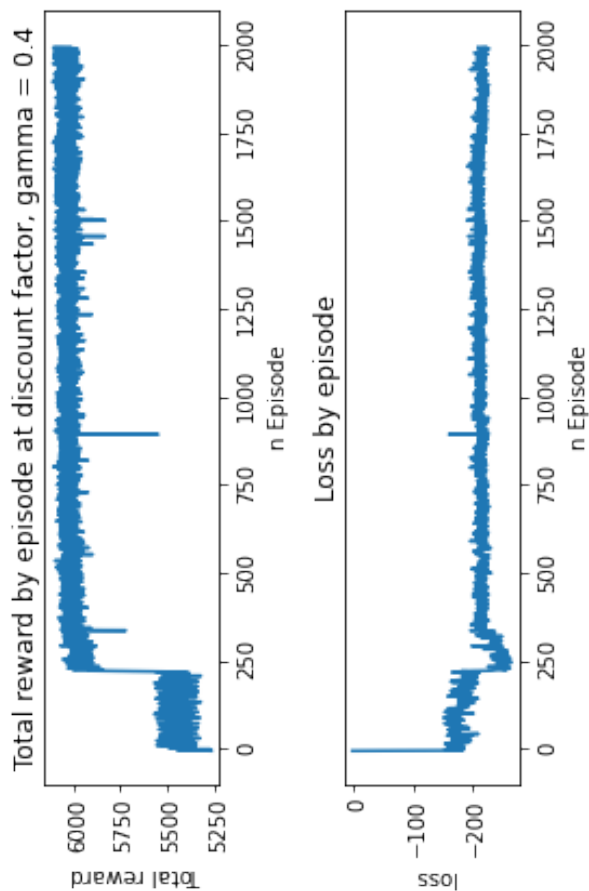


proposed approach presents significant improvement, which is represented by values of 9% and 3% at  $c = 10000$  for EMLP and SVM, respectively. At a high value of  $\eta$  ( $\eta = 50\%$ ), as indicated in Figure 5.15b, the approaches exhibit slight improvement when compared with Figure 5.15a. Again, the proposed approach significantly outperforms SOTA approaches by about 12% and 8% compared with EMLP and SVM, respectively, at same reference point of  $c=1000$ . Further, EMLP performs better than SVM at high values of  $c$ , as can be seen in Figures 5.15a and 5.15b, respectively.

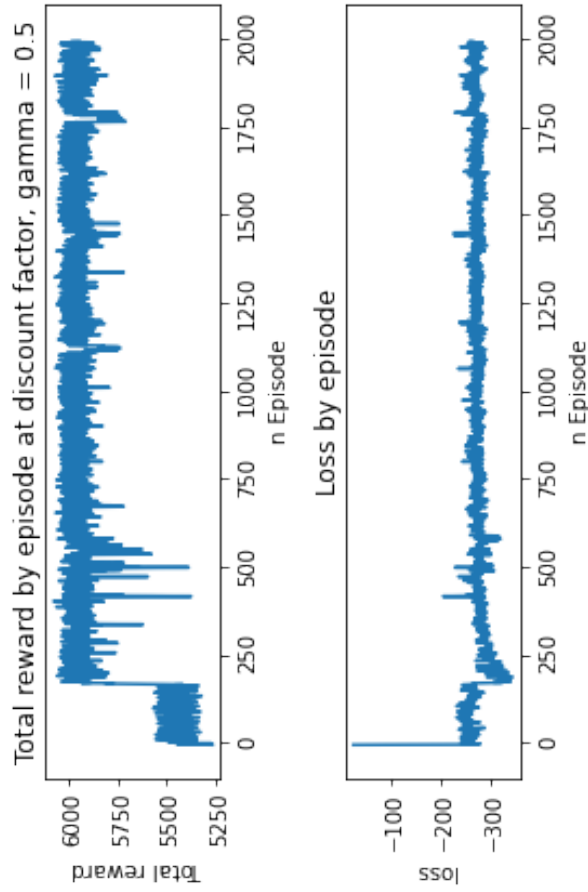
The proposed and SOTA approaches perform better as the value of  $\eta$  increases in the CAV network. This observation is aligned with expectation since smaller  $\eta$  values make the anomaly more challenging to detect.



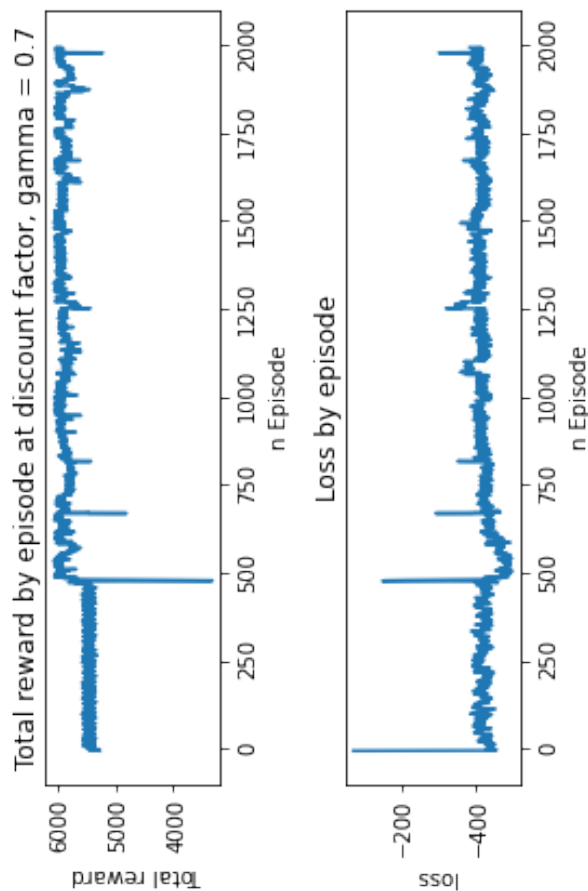
(A) Comparison of rewards and losses at different episodes of DDQN training at  $\gamma = 0.0$ .



(B) Comparison of rewards and losses at different episodes of DDQN training at  $\gamma = 0.4$ .  $\eta = 50\%$ .

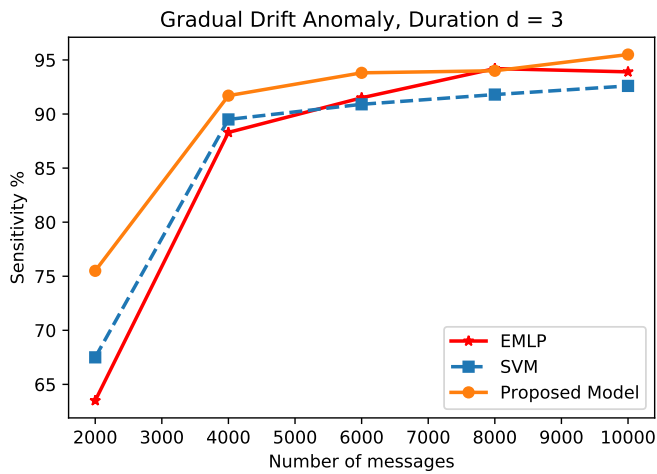


(C) Comparison of rewards and losses at different episodes of DDQN training at  $\gamma = 0.0$ .

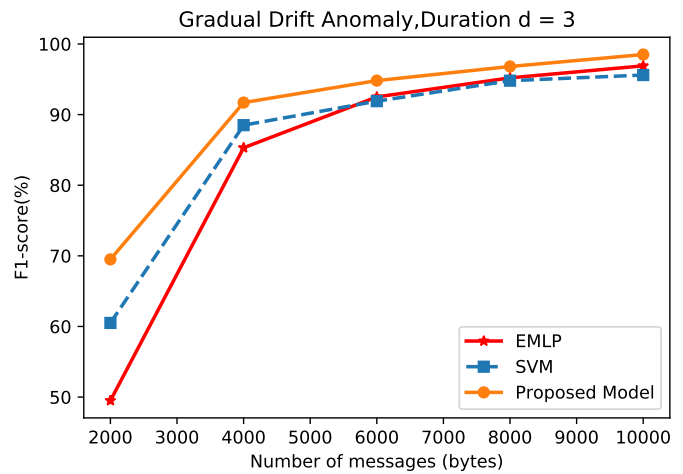


(D) Comparison of rewards and losses at different episodes of DDQN training at  $\gamma = 0.0$ .

FIGURE 5.13: Comparison of rewards and losses at different episodes of DDQN training at various  $\gamma$  values

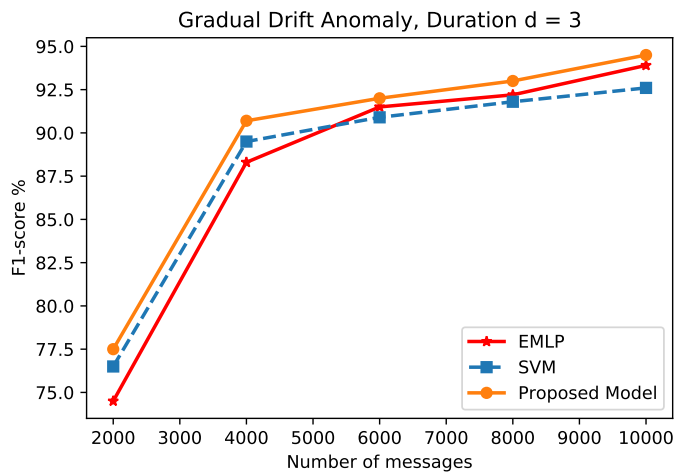


(A) Performance variation of the approaches in terms of  $F1 - score$  metric during the gradual drift anomaly scenario, at  $\eta=10\%$ .

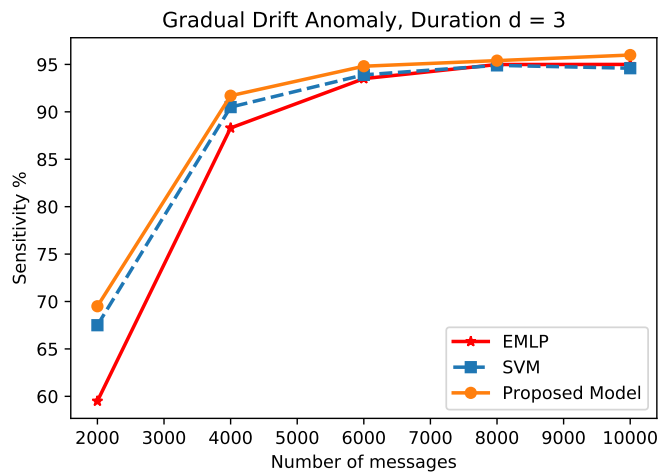


(B) Performance variation of the approaches in terms of  $F1 - score$  metric during the gradual drift anomaly scenario, at  $\eta =50\%$ .

FIGURE 5.14: Performance variation of the approaches during the gradual drift anomaly scenario, at  $\eta = 10\%$  and  $\eta = 50\%$



(A) Performance variation of the approaches in terms of sensitivity metric during the gradual drift anomaly scenario, at  $\eta =10\%$ .



(B) Performance variation of the approaches in terms of sensitivity metric during the gradual drift anomaly scenario, at  $\eta=10\%$ .

FIGURE 5.15: Performance variation of the approaches in terms of  $F1 - score$  and sensitivity metrics during the gradual drift anomaly scenario, at  $\eta = 10\%$  and  $\eta = 50\%$

The chapter demonstrates the performance of the detection approaches in the CAV and M2M networks. Results prove the effectiveness of ML/DL approaches in detecting abnormal sensor values (BSMs) in CAV and M2M networks. The proposed approaches such as BPSO-XGBoost, DWT-BDL, and DWT-DDQN provide better results than the SOTA.

# Conclusion and Future Work

---

## 6.1 Conclusion

CAVs are projected to transform today's transportation system into a highly efficient, automated, and intelligent one. In the near future, CAVs with varying levels of automation and connection are projected to cut travel time, enhance trip comfort, improve fuel efficiency, and minimize fatal accidents. Although expanding the use of CAV technology is expected to provide several benefits, it may also introduce difficulties in terms of safety, security, and privacy. In this context, recent records on crash events and successful automobile cyberattacks demonstrate that existing autonomous systems cannot manage unexpected complicated situations. This is why the CAV anomaly detection system's robustness is significant for the future.

This thesis proposes anomaly detection approaches to safeguard CAVs against abnormal sensor behavior or malicious cyberattacks. The proposed approaches with a combination of DWT with BDL and DDQN aim to detect anomalous information and denoise the sensor data in the CAV network. Specifically, simulations are carried out in two phases, one with a single anomaly and another with multiple anomaly scenarios. In the single anomaly

scenario, performances of the proposed anomaly detection approaches are compared with that of the SOTA for a single sensor data. The results show that the combined approaches improve on individual strengths. The results of the simulation indicate that the proposed DWT-BDL approach results in performance gains in terms of sensitivity, precision, and  $F1 - score$  during gradual drift anomaly, with improvements of 6.98%, 9.10%, and 7.37%, respectively, when compared with BDL and 11.89%, 7.32%, and 9.37%, respectively, when compared with CNN baseline approach.

Furthermore, the thesis proposes an innovative data-driven anomaly detection approach that combines DWT and DDQN methods in CAVs. The DDQN is modified to accommodate classification by taking the state as the data feature and the label as the action. The DWT and DDQN are combined to enhance anomaly detection performance and noise filtering in the CAV networks. The DWT smoothens the BSM sensor reading before the BSMs are fed into the DDQN approach.  $F1 - score$  and sensitivity are used to assess the performance of the proposed approach. As per the simulation results, the proposed DWT-BDL approach outperforms the baseline EMLP approach by about 20% and 10% and SVM approach by about 12% and 8% at low and high anomaly density distributions, respectively.

To facilitate detailed experiments on the effects of the anomaly parameters in the CAV setting, we use anomaly parameters such as the distribution, duration, and incident rate. Moreover, extensive simulations are run to investigate the effects of anomaly type, duration, and density of anomaly distribution. As presented in Section 5.2, results indicate that the performance of the different approaches generally improves with the increase in the magnitude of anomaly distribution and duration for different anomaly types, given that the density of anomaly is drawn from the same distribution.

In another work focused on the VBM2M-C setting, the VBM2M-C deals with modeling a scenario that depicts a vehicular communication setting that generates BSM features. In this setting, vehicles maintain inter-operability and reliable connection among themselves. The simulation involves some percentage of the vehicles being configured to communicate malicious messages for their benefit.

In this thesis, insight is drawn on the ML models' applicability in VBM2M-C in detecting abnormal behaviors. In this case, the proposed approach is a combination of BPSO and XGoost. The primary essence of integrating BPSO is to serve as an optimizer to the XGBoost approach. The proposed BPSO-XGBoost approach is applied to the generated BSM input vector, and the capability of detecting abnormal behavior in the network is evaluated. The ability of the proposed approach to capture anomalous activities are demonstrated in Section 5.1.

Further, the proposed approach's detection potential is evaluated, and the results of the combination of two characteristics are shown in Section 5.1. The *AR*, *TPR*, and *FPR* presented earlier in this thesis work show the performance values for the respective probability of attacks and the performance values of the individual detection approaches. The proposed method's performance shows significant improvements up to about 9%, 7% performance gain, and 2% lower than XGBoost and 4%, 9% performance gains, and 7% less compared to RF.

Furthermore, in the work, the performance of the proposed DWT-BDL is evaluated on gradual drift anomaly type, and the performance is compared with CNN-ALSTM and KF-CNN. The detection approaches' output is quantified in terms of F1-score, sensitivity and precision metrics.

At a high and low density of gradual drift anomaly distribution, the proposed approach demonstrates superior performance in  $F1 - score$  and sensitivity metrics with values of  $97 \pm 0.008$  and  $97.08 \pm 0.009$  and  $89.39 \pm 0.035$  and  $87.13 \pm 0.016$ .

#### **Limitation of the thesis**

1. The abnormal values of sensors used in the experiments are simulated and relies on the previous literature research, primarily because this form of data is not yet readily available. The tests are limited to onboard sensors due to the lack of data on CAVs.
2. Analysis of advanced computational complexity of the selected approaches and deployment to chips were not carried out in the work.
3. Additionally, the proposed DWT-BDL approach provides a better anomaly detection performance but has little complexity based on the training time. Though vehicle communication is driven by a more powerful computer that accommodates complexity.



## 6.2 Future Work

Emergency safety and robust and reliable CAV systems are key research fields for the coming years. The following items can extend and complement the research work presented in this thesis:

1. Distinguishing between anomalies and malicious information

In potential research, distinguishing between anomalous and malicious information can be beneficial, as this can affect the action taken to minimize the consequences. Identifying the type of anomaly that occurs can also be advantageous. The ability to figure out the type of anomaly will, in turn, allow certain real-life actions to be developed to minimize the impact of cyberattacks, thus contributing to the development of effective defensive mechanisms.

2. Computational complexity and communication overhead estimation

Estimating the memory and processor requirements for each detection approach's architecture will provide low communication overhead and guidance on which approach is more applicable in a resource-constrained environment like vehicles.

3. Addition of authentication schemes

Although authentication and encryption are outside this study's scope, all the network messages are believed to be authenticated. However, the CAV system can be enhanced with a practical, lightweight authentication scheme ideal for CAV networks, such as group signature or modified digital signature elliptical curve, to provide authentication of the sensor data to have a secure system.

---

## References

---

- [1] S. Ahmed and K. Tepe, "Entropy-based recommendation trust model for machine to machine communications," in *Ad Hoc Networks*, vol. 187, Springer, 2017, pp. 297–305.
- [2] N. Grace, P. Thornton, J. Johnson, K. Blythe, C. Oxley, C. Merrefield, I. Bartinique, D. Morin, R. Zhang, L. Johnson-Moffet, *et al.*, "Volpe center annual accomplishments: Advancing transportation innovation for the public good-january 2018," John A. Volpe National Transportation Systems Center (US), Tech. Rep., 2018.
- [3] J. den Hartog, N. Zannone, *et al.*, "Security and privacy for innovative automotive applications: A survey," *Computer Communications*, vol. 132, pp. 17–41, 2018.
- [4] S. Ahmed, "Trust establishment and management in adhoc networks," 2016.
- [5] J. Liu and A. J. Khattak, "Delivering improved alerts, warnings, and control assistance using basic safety messages transmitted between connected vehicles," *Transportation research part C: emerging technologies*, vol. 68, pp. 83–100, 2016.
- [6] R. Cai, Z. Zhang, A. K. Tung, C. Dai, and Z. Hao, "A general framework of hierarchical clustering and its applications," *Information Sciences*, vol. 272, pp. 29–48, 2014.

- [7] S. Ahmed, S. Al-Rubeaai, and K. Tepe, "Novel trust framework for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 9498–9511, 2017.
- [8] J.-H. Cho, A. Swami, and R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 562–583, 2011.
- [9] X. Liu, A. Datta, and E.-P. Lim, *Computational trust models and machine learning*. CRC Press, 2014.
- [10] M. Raya, P. Papaditos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, IEEE, 2008, pp. 1238–1246.
- [11] E. Eziam, K. Tepe, A. Balador, K. S. Nwizege, and L. M. Jaimes, "Malicious node detection in vehicular ad-hoc network using machine learning and deep learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*, IEEE, 2018, pp. 1–6.
- [12] F. Jahan, W. Sun, Q. Niyaz, and M. Alam, "Security modeling of autonomous systems: A survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–34, 2019.
- [13] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," *arXiv preprint arXiv:1810.00069*, 2018.
- [14] F. A. Ghaleb, A. Zainal, M. A. Rassam, and F. Mohammed, "An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications," in *Application, Information and Network Security (AINS), 2017 IEEE Conference on*, IEEE, 2017, pp. 13–18.

- [15] F. A. Ghaleb, A. Zainal, M. A. Maroof, M. A. Rassam, and F. Saeed, "Detecting bogus information attack in vehicular ad hoc network: A context-aware approach," *Procedia Computer Science*, vol. 163, pp. 180–189, 2019.
- [16] 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.
- [17] M. Raya, "Data-centric trust in ephemeral networks," 2009.
- [18] R. W. Van der Heijden, "Misbehavior detection in cooperative intelligent transport systems," Ph.D. dissertation, Universität Ulm, 2018.
- [19] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *International Conference on Security and Privacy in Communication Systems*, Springer, 2018, pp. 318–337.
- [20] W. Li, A. Joshi, and T. Finin, "Sat: An svm-based automated trust management system for mobile ad-hoc networks," in *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011*, IEEE, 2011, pp. 1102–1107.
- [21] Y. Wang, N. Masoud, and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [22] F. Van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1264–1276, 2019.
- [23] S. Mitra, S. Bose, S. S. Gupta, and A. Chattopadhyay, "Secure and tamper-resilient distributed ledger for data aggregation in autonomous vehicles,"

- in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, IEEE, 2018, pp. 548–551.
- [24] J. K. Choi and Y. G. Ji, “Investigating the importance of trust on adopting an autonomous vehicle,” *International Journal of Human-Computer Interaction*, vol. 31, no. 10, pp. 692–702, 2015.
- [25] D. Tokody, A. Albini, L. Ady, Z. Rajnai, and F. Pongracz, “Safety and security through the design of autonomous intelligent vehicle systems and intelligent infrastructure in the smart city,” *Interdisciplinary Description of Complex Systems: INDECS*, vol. 16, no. 3-A, pp. 384–396, 2018.
- [26] S. Parkinson, P. Ward, K. Wilson, and J. Miller, “Cyber threats facing autonomous and connected vehicles: Future challenges,” *IEEE transactions on intelligent transportation systems*, vol. 18, no. 11, pp. 2898–2915, 2017.
- [27] J. Petit and S. E. Shladover, “Potential cyberattacks on automated vehicles,” *IEEE Transactions on Intelligent transportation systems*, vol. 16, no. 2, pp. 546–556, 2014.
- [28] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, “Connected vehicles: Solutions and challenges,” *IEEE internet of things journal*, vol. 1, no. 4, pp. 289–299, 2014.
- [29] Z. Mahmood, “Connected vehicles in the iov: Concepts, technologies and architectures,” in *Connected Vehicles in the Internet of Things*, Springer, 2020, pp. 3–18.
- [30] R. Miucic, *Connected Vehicles: Intelligent Transportation Systems*. Springer, 2018.
- [31] M. Aloqaily, S. Otoum, I. Al Ridhawi, and Y. Jararweh, “An intrusion detection system for connected vehicles in smart cities,” *Ad Hoc Networks*, vol. 90, p. 101 842, 2019.

- [32] H. Khelifi, S. Luo, B. Nour, H. Moun gla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 320–351, 2019.
- [33] M. S. Sheikh and J. Liang, "A comprehensive survey on vanet security services in traffic management system," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.
- [34] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous vehicular networking: A survey on architecture, challenges, and solutions," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2377–2396, 2015.
- [35] R. Sabouni, "Evaluation of dsrc for v2v communications," Ph.D. dissertation, Carleton University, 2011.
- [36] X. Wu, S. Subramanian, R. Guha, R. G. White, J. Li, K. W. Lu, A. Buccheri, and T. Zhang, "Vehicular communications using dsrc: Challenges, enhancements, and evolution," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 399–408, 2013.
- [37] D. M. Vanderveen and K. Shukla, "Cellular v2x communications towards 5g," *5G Americas) White Paper*, 2018.
- [38] K. Z. Ghafoor, M. Guizani, L. Kong, H. S. Maghdid, and K. F. Jasim, "Enabling efficient coexistence of dsrc and c-v2x in vehicular networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 134–140, 2019.
- [39] A. Hegde and A. Festag, "Mode switching strategies in cellular-v2x," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 81–86, 2019.
- [40] A. Rawat, H. Shah, and V. Patil, "Towards intelligent vehicular networks: A machine learning framework," *Int. J. Res. Engin., Sci. Manage*, vol. 1, no. 9, pp. 2581–5782, 2018.

- [41] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, p. 16, 2018.
- [42] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, 2. MIT press Cambridge, 2016, vol. 1.
- [43] N. Kaja, "Artificial intelligence and cybersecurity: Building an automotive cybersecurity framework using machine learning algorithms," 2019.
- [44] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [45] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *arXiv preprint arXiv:1906.05799*, 2019.
- [46] B. Sheehan, F. Murphy, M. Mullins, and C. Ryan, "Connected and autonomous vehicles: A cyber-risk classification framework," *Transportation research part A: policy and practice*, vol. 124, pp. 523–536, 2019.
- [47] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.
- [48] J. Petit, M. Feiri, and F. Kargl, "Spoofed data detection in vanets using dynamic thresholds," in *Vehicular Networking Conference (VNC), 2011 IEEE*, IEEE, 2011, pp. 25–32.
- [49] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, *et al.*, "Comprehensive experimental analyses of automotive attack surfaces.," in *USENIX Security Symposium*, San Francisco, vol. 4, 2011, pp. 447–462.

- [50] A. Weimerskirch and R. Gaynier, "An overview of automotive cybersecurity: Challenges and solution approaches.," in *TrustED@ CCS*, 2015, p. 53.
- [51] K. Salahshoor, M. Mosallaei, and M. Bayat, "Centralized and decentralized process and sensor fault monitoring using data fusion based on adaptive extended kalman filter algorithm," *Measurement*, vol. 41, no. 10, pp. 1059–1076, 2008.
- [52] M. Sewak and S. Singh, "Iot and distributed machine learning powered optimal state recommender solution," in *2016 International Conference on Internet of Things and Applications (IOTA)*, IEEE, 2016, pp. 101–106.
- [53] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 1110–1115.
- [54] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, IEEE, 2016, pp. 1–6.
- [55] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.
- [56] B. Škorić, S. J. de Hoogh, and N. Zannone, "Flow-based reputation with uncertainty: Evidence-based subjective logic," *International Journal of Information Security*, vol. 15, no. 4, pp. 381–402, 2016.
- [57] P. Godsmark, B. Kirk, V. Gill, and B. Flemming, "Automated vehicles: The coming of the next disruptive technology," 2015.
- [58] Y. Yang, Q. Feng, Y. L. Sun, and Y. Dai, "Reptrap: A novel attack on feedback-based reputation systems," in *Proceedings of the 4th international*



- conference on Security and privacy in communication networks*, ACM, 2008, p. 8.
- [59] A. Petrillo, A. Pescape, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE transactions on cybernetics*, vol. 51, no. 3, pp. 1134–1149, 2020.
- [60] J. den Hartog, N. Zannone, *et al.*, "Feature selection for anomaly detection in vehicular ad hoc networks," in *15th International Joint Conference on e-Business and Telecommunications, ICETE 2018*, SCITEPRESS-Science and Technology Publications, Lda., 2018, pp. 481–491.
- [61] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [62] H. Ji, Y. Wang, H. Qin, Y. Wang, and H. Li, "Comparative performance evaluation of intrusion detection methods for in-vehicle networks," *IEEE Access*, vol. 6, pp. 37 523–37 532, 2018.
- [63] E Ruffio, D Saury, D Petit, and M Girault, "Tutorial 2: Zero-order optimization algorithms," *Eurotherm School METTI*, 2011.
- [64] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.
- [65] B. H. Nguyen, B. Xue, P. Andreae, and M. Zhang, "A new binary particle swarm optimization approach: Momentum and dynamic balance between exploration and exploitation," *IEEE transactions on cybernetics*, vol. 51, pp. 589 –603, 2019.
- [66] R. Sseguya, *Forecasting anomalies in time series data from online production environments*, 2020.

- [67] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*, 10. Springer series in statistics New York, 2001, vol. 1.
- [68] S. K. Kiangala and Z. Wang, "An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-xgboost and random forest ensemble learning algorithms in an industry 4.0 environment," *Machine Learning with Applications*, vol. 4, p. 100 024, 2021.
- [69] S. Misra and H. Li, "Noninvasive fracture characterization based on the classification of sonic wave travel times," *Machine Learning for Subsurface Characterization*, pp. 243–287, 2019.
- [70] M. Chen, Q. Liu, S. Chen, Y. Liu, C.-H. Zhang, and R. Liu, "Xgboost-based algorithm interpretation and application on post-fault transient stability status prediction of power system," *IEEE Access*, vol. 7, pp. 13 149–13 158, 2019.
- [71] Y. Gu, D. Zhang, and Z. Bao, "A new data-driven predictor, pso-xgboost, used for permeability of tight sandstone reservoirs: A case study of member of chang 4+ 5, western jiyuan oilfield, ordos basin," *Journal of Petroleum Science and Engineering*, vol. 199, p. 108 350, 2021.
- [72] M. Li, Z. Wang, J. Luo, Y. Liu, and S. Cai, "Wavelet denoising of vehicle platform vibration signal based on threshold neural network," *Shock and Vibration*, vol. 2017, 2017.
- [73] C. Bishop, C. M. Bishop, *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.
- [74] S Arangio and J. Beck, "Bayesian neural networks for bridge integrity assessment," *Structural Control and Health Monitoring*, vol. 19, no. 1, pp. 3–21, 2012.
- [75] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, 2016.

- [76] Z. Ghahramani, "A history of bayesian neural networks," in *NIPS Workshop on Bayesian Deep Learning*, 2016.
- [77] H. S. Rodrigo, *Bayesian artificial neural networks in health and cybersecurity*. University of South Florida, 2017.
- [78] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, p. 112 963, 2020.
- [79] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.
- [80] S. T. Goh, "Machine learning approaches to challenging problems: Interpretable imbalanced classification, interpretable density estimation, and causal inference," Ph.D. dissertation, Massachusetts Institute of Technology, 2018.
- [81] O. O. Koyejo, N. Natarajan, P. K. Ravikumar, and I. S. Dhillon, "Consistent binary classification with generalized performance metrics," in *Advances in Neural Information Processing Systems*, 2014, pp. 2744–2752.
- [82] D. Silvestro and T. Andermann, "Prior choice affects ability of bayesian neural networks to identify unknowns," *arXiv preprint arXiv:2005.04987*, 2020.
- [83] A. Maier, B. Lorch, and C. Riess, "Toward reliable models for authenticating multimedia content: Detecting resampling artifacts with bayesian neural networks," *arXiv preprint arXiv:2007.14132*, 2020.
- [84] A. R. Javed, M. Usman, S. U. Rehman, M. U. Khan, and M. S. Haghghi, "Anomaly detection in automated vehicles using multistage attention-based convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, 2020.


- [85] E. Eziama, F. Awin, S. Ahmed, L. Marina Santos-Jaimes, A. Pelumi, and D. Corral-De-Witt, "Detection and identification of malicious cyber-attacks in connected and automated vehicles' real-time sensors," *Applied Sciences*, vol. 10, no. 21, p. 7833, 2020.
- [86] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *arXiv preprint arXiv:2103.05127*, 2021.
- [87] D. Laredo, S. F. Ma, G. Leylaz, O. Schütze, and J.-Q. Sun, "Automatic model selection for fully connected neural networks," *International Journal of Dynamics and Control*, vol. 8, no. 4, pp. 1063–1079, 2020.
- [88] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive neural architecture search," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 19–34.
- [89] J. Guo, X. Li, Z. Liu, J. Ma, C. Yang, J. Zhang, and D. Wu, "Trove: A context-awareness trust model for vanets using reinforcement learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6647–6662, 2020.
- [90] X. Larriva-Novo, M. Vega-Barbas, V. A. Villagra, D. Rivera, M. Alvarez-Campana, and J. Berrocal, "Efficient distributed preprocessing model for machine learning-based anomaly detection over large-scale cybersecurity datasets," *Applied Sciences*, vol. 10, no. 10, p. 3430, 2020.
- [91] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.







## Appendix A

---

# Copyright Permissions

---

Request for your permission to refer our publication in my Ph.D. Thesis 

 Luz Marina Santos Jaimes <lsantos@unipamplona.edu.co>  
Mon 2020-11-09 7:43 AM  
To: Ugonna Ezianya     

Dear Elvin,


I am giving my consent to refer to our publication in your Ph.D. thesis work.

Article:  
Detection and Identification of Malicious Cyber-Attacks in Connected and Automated Vehicles' Real-Time Sensors.

Journal:  
2020 MDPI Journal of applied science, Electrical, Electronic and Communications Engineering, Basel, Switzerland.

Regards

Luz Marina Santos Jaimes  
Ingeniería de Sistemas  
Grupo de Investigación Ciencias Computacionales  
Universidad de Pamplona, Colombia



[Reply](#) | [Forward](#)

Request for your permission to refer our publication in my Ph.D. Thesis

Sabbir Ahmed

...

SA

Saneeha Ahmed <saneeha.ahmed@algomau.ca>

Fri 2020-11-06 2:36 PM

To: Ugonna Eziana

Cc: Kemal Tepe

    ...

Dear Elvin,  
[Congratulations](#) for reaching this milestone. Yes, of course you can use this work in your thesis.  
Regards  
Saneeha

...

UE

Ugonna Eziana

Fri 2020-11-06 2:02 PM

To: Kemal Tepe

Bcc: Sabbir1009@yahoo.com <sabbir1009@yahoo.com>; Farooq Awin <awin@uwingmail.uwindsor.ca>; saneeha.ahmed@algomau.ca

    ...

Dear Co-authors,

I hope everyone is doing great. I am writing this email to seek your permission to refer to our publication in my Ph.D. thesis work.  
Article:  
Detection of Adversary Nodes in Machine-To-Machine Communication Using Machine Learning Based Trust Model. ISSPIT 2019: 1-6  
Conference:  
IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2019, Ajman, United Arab Emirates, December 10-12, 2019. IEEE 2019, ISBN 978-1-7281-5341-4

Request for your permission to refer our publication in my Ph.D. Thesis

TH LUVH,  
You have my permission.  
Regards  
Sabbir

Sabbir Ahmed

...



Translate message to: English | Never translate from: Spanish

D

Danilo Corral De Witt <dancowi@yahoo.com>

Fri 2020-11-06 2:35 PM

To: Ugonna Eziana

Authorized.

**Danilo Corral De Witt**  
+1 226 246 5278

    ...

UE

Ugonna Eziana

Fri 2020-11-06 2:19 PM

To: Danilo Corral De Witt <dancowi@yahoo.com>

Bcc: Sabbir1009@yahoo.com <sabbir1009@yahoo.com>; Luz Marina Santos Jaimes <lsantos@unipamplona.edu.co> +2 others

    ...

Dear Co-authors,

I hope everyone is doing great. I am writing this email to seek your permission to refer to our publication in my Ph.D. thesis work.

Article:

Request for your permission to refer our publication in my Ph.D. Thesis

Sabbir Ahmed

...

SA

Saneeha Ahmed <saneeha.ahmed@algomau.ca>

Fri 2020-11-06 2:36 PM

To: Ugonna Eziana

Cc: Kemal Tepe

Dear Elvin,

[Congratulations](#) for reaching this milestone. Yes, of course you can use this work in your thesis.

Regards

Saneeha

...

UE

Ugonna Eziana

Fri 2020-11-06 2:02 PM

To: Kemal Tepe

Bcc: Sabbir1009@yahoo.com <sabbir1009@yahoo.com>; Farooq Awin <awin@uwingmail.uwindsor.ca>; saneeha.ahmed@algomau.ca

Dear Co-authors,

I hope everyone is doing great. I am writing this email to seek your permission to refer to our publication in my Ph.D. thesis work.

Article:

Detection of Adversary Nodes in Machine-To-Machine Communication Using Machine Learning Based Trust Model. ISSPIT 2019: 1-6

Conference:

IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2019, Ajman, United Arab Emirates, 2019-11-10-13, 2019, IEEE, 2019, ISBN: 978-1-7994-2344-4

Request for your permission to refer our publication in my Ph.D. Thesis

[Reply](#) | [Reply all](#) | [Forward](#)

JA

James Agajo <agajojul@gmail.com>

Sun 2020-11-08 9:53 AM

To: Ugonna Eziana

Cc: Kemal Tepe; s.k.nwizege@ieee.org; ali.balador@mdh.se; Luz Marina Santos Jaimes <lsantos@unipamplona.edu.co>

Yes, go ahead.

...

AB

Ali Balador <ali.balador@mdh.se>

Fri 2020-11-06 4:46 PM

To: Ugonna Eziana

Dear Elvin,

It is fine with me. Please go ahead.

Best regards,

Ali Balador.

Get [Outlook for iOS](#)

...

UE

Ugonna Eziana

Fri 2020-11-06 2:26 PM

To: Kemal Tepe

Cc: s.k.nwizege@ieee.org; ali.balador@mdh.se; agajojul@gmail.com; Luz Marina Santos Jaimes <lsantos@unipamplona.edu.co>

**Request for your permission to refer our publication in my Ph.D. Thesis**

**FA** Faroq Awin  
Yes, I approve. Fri 2020-11-06 9:55 PM

---

**SA** Sabbir Ahmed <sabbir1009@yahoo.com>  
Fri 2020-11-06 9:05 PM  
To: Kemal Tepe; Ugonna Eziana  
Hi Elvin,  
You have my permission.  
Regards  
Sabbir  
  
Sabbir Ahmed  
...

---

**SA** Saneeha Ahmed <saneeha.ahmed@algonau.ca>  
Fri 2020-11-06 2:36 PM  
To: Ugonna Eziana  
Cc: Kemal Tepe  
Dear Elvin,  
**Congratulations** for reaching this milestone. Yes, of course you can use this work in your thesis.  
Regards  
Saneeha  
...

---

**UE** Ugonna Eziana  
Fri 2020-11-06 2:02 PM  
To: Kemal Tepe

**Request for your permission to refer our publication in my Ph.D. Thesis**

[Reply](#) | [Reply all](#) | [Forward](#)

---

**JA** James Agajo <agajojul@gmail.com>  
Sun 2020-11-08 9:53 AM  
To: Ugonna Eziana  
Cc: Kemal Tepe; s.k.nwizege@ieee.org; ali.balador@mdh.se; Luz Marina Santos Jaimes <lsantos@unipamplona.edu.co>  
Yes, go ahead.  
...

---

**AB** Ali Balador <ali.balador@mdh.se>  
Fri 2020-11-06 4:46 PM  
To: Ugonna Eziana  
Dear Elvin,  
  
It is fine with me. Please go ahead.  
  
Best regards,  
Ali Balador.  
  
Get [Outlook for iOS](#)  
...

---

**UE** Ugonna Eziana  
Fri 2020-11-06 2:26 PM  
To: Kemal Tepe

**Request for your permission to refer our publication in my Ph.D. Thesis**

**PP** pelumi pelumi <akinyemiakinpelumi@gmail.com>  
Sat 2020-11-07 1:47 AM  
To: Ugonna Eziana

Dear Mr. Eziana,  
I am pleased to inform you that your request to refer the journal title: Detection and Identification of Malicious Cyber-Attacks in Connected and Automated Vehicles' Real-Time Sensors in your PhD thesis have been granted

Akinyemi, PhD

Sent from my iPhone

On Nov 6, 2020, at 2:19 PM, Ugonna Eziana <eziama@uwindsor.ca> wrote:

Elvin Eziana

[Reply](#) | [Forward](#)



## Mathematical Notations

Ord	Variable	Unit and /or Observations
1	$S(n)$	Original signal / dBm
2	$m_{j,k}$	Wavelet coefficient as a result of wavelet decomposition
3	$E_D$	Error between the data
4	$N_o$	Number of output variables
5	$N_m$	Number of candidates
6	$\alpha$	Optimization parameter
7	$\eta$	Incident/anomaly rate
8	$\mathcal{N}$	Normal distribution
9	$\mathcal{U}$	Uniform distribution
10	$d$	Anomaly duration/(s)
11	$\mu$	Mean
12	$m$	Vehicular density
13	$\sigma$	Standard deviation
14	$f(n)$	Estimated wavelet coefficient
15	$d_{i,k}$	Detailed coefficient
16	$c_{j,k}$	Approximate coefficient
17	$\varepsilon(t)$	Estimated wavelet
18	$T(V_{i,j})$	Transfer function
19	$v_i$	Velocity m/s
20	$q$	Percentage of malicious information / %
21	$A_x$	Lateral acceleration
22	$\beta$	Optimization parameter

# Mathematical Derivations and Illustrations

---

### Bayesian classification with Three Variables

The prior distribution of the parameter(s) before any observation is experienced is given by  $p(C | D)$ . This prior distribution might not be determined easily; in such case Jeffery's prior is used to get prior distribution before the update of a newer observation. The sampling distribution or likelihood is the distribution of the observed data based on condition of its parameters, i.e.  $p(M_j | C)$  The marginal likelihood  $p(M_j | D) = \int p(M_j | C, D) dC$

The posterior distribution is the distribution of the parameter(s) after having considered of the observed data.

$$\begin{aligned} p(C | M_j, D) &= \frac{p(C, M_j, D)}{p(M_j, D)} \\ &= \frac{p(M_j | C, D) p(C, D)}{p(M_j, D)} \\ &= \frac{p(M_j | C, D) p(C | D) p(D)}{p(M_j, D)} \\ &= \frac{p(M_j | C, D) p(C | D) p(D)}{p(M_j | D) p(D)} \end{aligned}$$

$$= \frac{p(M_j|C,D)p(C|D)}{p(M_j|D)} \propto p(M_j | C, D) p(C | D)$$

### Particle Swarm Optimization position and velocity

Mathematically, the particle's position and velocity are expressed as follows:

$$X_i = (X_{i1}, X_{i2}, \dots, X_{ik}), X_{ij} \in \{0, 1\}, j = 1, 2, 3, \dots, K$$

$$V_i = (V_{i1}, V_{i2}, \dots, V_{ik}), V_{ij} \in \{-V_{max}, V_{max}\}, j = 1, 2, 3, \dots, K$$

where  $X_i$  position represents a solution, with an associated fitness value.

The velocity function of  $t + 1$  iteration is represented as follows:  
 $V_i^{t+1} = wV_i^t + C_1r_1 (P_i - X_i^t + C_2r_2) (P_g - X_i^t)$  where  $P_i, P_g$  denote the best position visited by particle  $i$  and the best position found by the swarm,  $w$  is an inertia factor that varies over time, and  $r_1, r_2$  are the random values uniform on  $[0, 1]$ . The transfer function denoted by  $T(V_{ij})$  helps in converting velocities to probabilities and it is expressed as follows:

$$T(V_{ij}) = \frac{1}{1 + e^{V_{ij}}} \quad (C.1)$$

this transfer function equally helps in updating each bit position as follows:

$$X_{ij} = \begin{cases} 1 & \text{if } \cup(0, 1) < T(V_{ij}) \\ 0 & \text{otherwise} \end{cases} \quad (C.2)$$

## Appendix D

# Key Literature Related to Anomaly Detection Approaches in CAV

Author(s)	Cyber-security/anomaly aspects investigated	key relevant findings	Study approach	Type of data
Wang, Y.; Masoud N.; Khojandi, A.	Anomaly detection and recovery in connected automated vehicle sensors.	Identifies faults/ anomalies in CAV networks. Detection approach able to detect types of anomalies/ attacks. Approach is critically affected by uncertainty in noise processing	Observed-based approach using AEKF with SVM	BSM data-set
Van Wyk, F.; Wang, Y.; Khojandi, A.; Masoud, N.	Detection and identification of anomaly in automated vehicles	Developed a methodology that can seamlessly detect anomalies and their source in real time.	CNN with kalman Filter based approach.	BSM data-set
Godsmark, P.; Kirk, B.; Gill, V. ; Flemming, B.	Deployment of smart protection system to secure the external communication of self driving cars .	Detection of both grey hole and rushing attack using the proposed approach.	SVM and FFNN based approach	Field experiment using a vehicle CAN data
Petit and Shladover	State of the art in identifying potential cyber attacks on automated vehicles	Identifies risks of various importance in identifying GNSS spoofing and fake message injection as the most dangerous attack on AVs	Exploratory study	Literature review
Muter and Asaj	Detecting anomalies /attacks for in-vehicle networks	Certain attacks on the CAN-bus of a vehicle were detected using the proposed methodology. Difficult to detect low - volume attacks in CAN-bus of vehicles	Signal Entropy	Field experiment using a vehicle CAN data
Marchett	Detection of anomalous/ attacks for in-vehicle networks	Detection of attacks on CAN-bus of vehicles	Signal Entropy	Field experiment with a vehicle CAN data
Weimerskirch, A.; Gaynier, R.	Overview of automatic cybersecurity, challenges and solution approach	Identifies the cyber challenges and possible solutions	Exploratory study	Review of literature
Petrillo, A.; Pescape, A.; Santini, S.	Addresses the problem of cyber threat for a vehicle platoon	Efficiency of adaptive synchronization on the basis of control algorithm to mitigate adversary information. Approach not scalable in high density network	Lyapunov-Krasovski theory	Simulation using PLEXE

---

## **Vita Auctoris**

---

Elvin Ezianya received his Bachelors and Masters degree from University of Nigeria Nsukka and Federal University of Technology Owerri, Nigeria, in 2006 and 2012, respectively. During his graduate study, he taught various Communication Engineering courses, such as Electromagnetic Waves and Antenna. His research interests are emergency in connected and automated vehicles (CAVs), Machine to Machine (M2M) and Deep Learning. His research focused on improving the safety of CAVs under cyber-security and safety uncertainties.