International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/224442 Volume 11, Issue 2, March – April (2024)

RESEARCH ARTICLE

Machine Learning Based Misbehavior Detection System for False Data Injection Attack in Internet of Vehicles Using Neighbor Public Transport Vehicle Approach

Hussaini Aliyu Idris
Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology (E-JUST), New Borg-El-Arab City, Alexandria, Egypt.
hussaini.idris@ejust.edu.eg

Kazunori Ueda
Department of Computer Science and Engineering, Waseda University Tokyo, Japan.
ueda@ueda.info.waseda.ac.jp

Bassem Mokhtar
College of Information Technology, UAE University, Al Ain, UAE.
bassem.mokhtar@uaeu.ac.ae

Samir A. Elsagheer Mohamed
Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology (E-JUST)
New Borg-El-Arab City, Alexandria, Egypt.
samir.elsagheer@ejust.edu.eg

Received: 15 December 2023 / Revised: 15 March 2024 / Accepted: 23 March 2024 / Published: 30 April 2024

Abstract – The integration of the Internet of Vehicles (IoV) into the Intelligent Transportation System (ITS) has significantly improved its operations, leading to a reduction in road traffic accidents, efficient traffic control, and a decrease in carbon emissions for a more sustainable environment aligned with the Sustainable Development Goals (SDGs). However, the adoption of IoV networks introduces privacy and security challenges. Although cryptographic techniques such as public-key infrastructure (PKI) proposed by standardization bodies like IEEE and ETSI provide protection against outsider attackers, they fail to address the threat posed by insider attackers. To overcome this limitation, researchers have proposed data-centric machine learning-based misbehavior detection frameworks that focus on identifying and mitigating insider attacks. However, existing approaches primarily rely on the Basic Safety Message (BSM) data received from a single vehicle, which allows attackers to manipulate the BSM data without being detected. In this paper, we present a novel data-centric misbehavior detection framework specifically designed to detect false data injection attacks in IoV networks. Our approach leverages neighboring public transportation vehicles (NPTVs) to enhance the detection capabilities. By incorporating the BSM data from NPTVs, we demonstrate the effectiveness of our proposed framework in different scenarios using deep learning, decision tree, and random forest algorithms. Through extensive evaluation, we achieved precision, recall, F1-Score, and accuracy rates of up to 99%, showcasing the superior performance of our approach.

Index Terms – Machine Learning, Internet of Vehicles, Misbehavior Detection System, Intrusion Detection, Intelligent Transportation System, Basic Safety Message.

1. INTRODUCTION

As countries worldwide strive to combat road traffic accidents and set ambitious targets of achieving zero traffic fatalities by 2050 [1], it becomes imperative for the transportation system to leverage cutting-edge technologies in conjunction with stringent laws and policies to realize this goal. In response to this need, the Intelligent Transportation System (ITS) has emerged, with the Internet of Vehicles (IoV) serving as the foundational framework. The overarching objectives of the ITS encompass accident reduction, traffic management, optimal road utilization, substantial reduction in vehicle emissions to mitigate air pollution, and enhanced passenger experience [2].

The IoV plays a pivotal role in enabling the comprehensive functionality of the ITS by facilitating various advanced forms of communication. These include intra-vehicle communication such as vehicle-to-sensors (V2S); and inter-vehicle communication or vehicle-to-vehicle (V2V), as well as other types of communication, such as vehicle-to-infrastructure (V2I), vehicle-to-cloud (V2C), vehicle-to-pedestrian (V2P), and vehicle-to-smart-electric-grid (V2G). Collectively referred to as vehicle-to-everything (V2X) communication, these mechanisms empower connected and autonomous vehicles (CAVs) to exchange crucial information with their surrounding environments, thus promoting improved traffic management and monitoring capabilities[3], [4].

Wireless communications within the IoV environment are achieved through the utilization of the IEEE 802.11P standard, an amendment to the IEEE 802.11 (Wi-Fi standard). This standard gave rise to the IEEE 1609 family, which encompasses protocols such as Wireless Access in Vehicular Environments (WAVE) and Dedicated Short-Range Communication (DSRC). Alternatively, IoV communications can also take place via cellular mobile networks, including 4G/LTE, 5G, and future iterations [5], [6].

The intricate nature of communication within the Intelligent Transportation System (ITS) and the Internet of Vehicles (IoV) renders the latter susceptible to cyberattacks. Given that vehicles transmit vital basic safety messages (BSMs) over vehicle-to-everything (V2X) communication channels for the protection of drivers, passengers, and vulnerable road users (VRUs), proactive security measures have been established to safeguard against malicious attackers attempting to tamper with the content of these BSMs.

1.1. Problem Statement

To fortify the cooperative intelligent transportation system (c-ITS) against security threats, cryptographic techniques employing public-key-infrastructure (PKI) have been implemented. This cryptographic framework is designed to address critical security requirements encompassing authentication, confidentiality, integrity, availability, non-repudiation, and privacy [7].

However, while these cryptographic security measures effectively shield the IoV from external attackers, they do not adequately tackle the potential risks posed by insider threats. Consequently, there arises a pressing need for a data-centric misbehavior detection system that can intelligently identify and flag any aberrant behavior exhibited by authenticated vehicles within the network.

1.2. Motivation

The research community has made significant contributions to proposing diverse data-centric machine learning-based approaches for the detection and mitigation of cyberattacks and other types of anomalous behavior displayed by malicious attackers within IoV networks. However, existing approaches in the literature heavily rely on the data (features) of individual BSMs for machine learning training and inference, thereby allowing a malicious attacker to effectively manipulate BSMs to evade detection. These approaches fail to emulate the cooperative nature of the ITS environment, resulting in a high rate of false negative and false positive alarms. This research proposes a novel misbehavior detection system (MDS) by utilizing neighboring public transport vehicles.

1.3. Objective

It is important to note that the primary contribution of this paper lies not in designing a novel machine learning algorithm but in introducing the concept of utilizing combined features from BSMs of two neighboring vehicles, one of which being a public transport vehicle, to train machine learning classifiers. The proposed approach demonstrates superior performance compared to existing methods, even when utilizing the same dataset and some similar machine learning algorithms. The key contributions of this study are as follows:

- We propose a novel, computationally efficient, and robust data-centric machine learning-based framework that utilizes BSM data from licensed neighboring public transportation vehicles (NPTVs), such as public transport buses and trams, to detect false data injection attacks in the IoV network.
- Generate a novel vehicular misbehavior dataset based on NPTV from the original Burwood SUMO Traffic Australian Dataset for Misbehavior Detection (BurST-ADMA) dataset
- Furthermore, our proposed approach investigates distinct scenarios based on the number of NPTVs present at a given time instance.
- Finally, we conduct a comparative study involving several misbehavior detection frameworks from the literature, utilizing the BurST-ADMA dataset, to assess the effectiveness of our proposed framework.

1.4. Organization of the Paper

The remainder of this paper is organized as follows: Section 2 provides a comprehensive review of related work. Section 3 presents our proposed data-centric machine learning-based framework for detecting false data injection attacks in the IoV. In Section 4, we discuss the simulation studies conducted and present the obtained results. The paper concludes with a discussion on future research directions in Section 5.

2. RELATED WORK

The development of a data-centric misbehavior detection framework within the Intelligent Transportation System (ITS) has garnered significant attention within the research community. Various scholars have made valuable contributions to this field, as evidenced in recent surveys, articles and review publications such as those by the authors in [8], [9], [10], [11]and [12]. However, it is important to note that these existing frameworks predominantly rely on the data of a single vehicle's BSM, which may include data from potential attackers, for both training and prediction purposes. This approach inadvertently grants malicious attackers' complete freedom to manipulate falsified Basic Safety Messages (BSMs). In this section, we aim to provide an overview of some of the state-of-the-art works available in the literature and conduct a comparative analysis with our own work.

The authors in [13] presented a supervised learning- based MDS to identify position falsification attacks by changing the VeReMi dataset [14] to produce another dataset comprising two successive BSMs features for ML training. In comparison to earlier research, they trained binary classifiers and multi-class classifiers utilizing decision tree, logistic regression, random forest, K-Nearest Neighbor, and Naive bayes machine learning (ML) algorithms to achieve superior performance using recall, F1-Score, and precision measures. The proposed MDS was designed to be installed on RSUs with a shared database for storing and retrieving consecutive BSMs. While this approach recorded significant performance, it made the assumption that internal attacker always disseminates falsified BSMs which is incorrect in real-world scenario. A location spoofing attack was detected in a similar effort by the same authors in [15].

Moreover, the authors in [16] presented a machine learning based misbehavior detection system to identify position falsification attacks using VeReMi dataset. The approach utilized with 20 features from BSM data based on position differences between sending vehicles and receiving vehicles. The limitation of this approach is in the obtained result and computation for the differences in position between the vehicles tend to be expensive for time-sensitive IoV environment.

In addition, the authors in [17] proposes a data-centric machine learning approach by integrating six supervised ML algorithms with plausibility checks against the normal behavior (data) of benevolent vehicles in the network before passing to ML for final prediction. The approach demonstrates the effect of plausibility checks on the raw BSM data with the only shortcoming being relatively low precision and recall, implying high rate of false alarm.

Further, the authors in [18] proposed a supervised learning approach in which support vector machine (SVM) and logistic regression (LR) classifiers with and without normalization are employed. Feature selection of x,y,z speed and position coordinates were used as the selected features for training the classifiers. Even though SVM with normalization outperformed the rest of the used classifiers, the obtained accuracy of a little over 96% need further improvement to suit IoV.

Furthermore, the authors in [19] proposed a broad learning system (BLS) that takes raw BSM data from the vehicles and extracts six critical features: vehicle speed, position, heading, transmission delay, broadcast speed, and vehicle acceleration. A deep learning algorithm, specifically stacked long short-term memory (LSTM) recurrent neural network (RNN) is trained using these features. While the approach recorded a superior performance, it is only suitable for edge servers deployed near roadside units (RSUs) for its computational requirements.

Moreover, the authors presented a novel hyperparameter-tuned ensemble Random Forest (RF) classifier based on majority voting in [20] to detect bogus basic safety messages (BSM) in IoV. The RF was hyper-tuned by iterative process of selection among an array of hyper-parameters for training using the VeReMi dataset. The approached achieved great result but the model-centric approach of the approach might not generalize well with other datasets or real-world data.

Similar to the prior work, the authors in [21] suggested an approach based on Randomized Search Optimization (RSO) that used majority voting to train an ensemble of random forest to identify fake BSM transmission by suspicious vehicles in the IoV network using BurST-ADMA dataset. Similar to the previous work, the authors in [19] proposed an ensemble of AdaBoost that uses weighted majority vote on several weak predictions to detect false messages in IoV using BurST-ADMA dataset and evaluated and compared their results in terms of accuracy, recall, precision, and F-measure metrics. The limitation of those approaches is that they are dataset-specific approach with tendency of failing against real-world data.

The approach in [22] uses supervised machine learning to detect false data injection attack. BSM data is collected form vehicles in traffic based on IoV context before undergoing preprocessing stages. The proposed approach used NGSIM dataset. The limitation of this approach is the modification of dataset to contain attacks by introducing noise by the authors hence repeatability and comparison become an issue.

The authors in [23] proposed a scheme that uses (i) a deep learning binary classification model deployed at the RSU edge server to detect message trustworthiness based on the vehicle dependability score (VDS) assigned by the Trusted Authority (TA) at a time when a vehicle joined the network and (ii) a graph temporal network (GTN) with attention to detect potentially malicious vehicles based on time sequential data. The shortcoming of this approach is added overhead for assigning dependability score.

The authors of the work in [24] offered two classification models for binary classification of sequence data supplied by the network's vehicles. Stacked LSTM and convolutional neural network (CNN)-LSTM models are trained, and CNN-LSTM finally outperforms stacked LSTM. DeepADV, the suggested framework, takes a succession of signals classified as legitimate, attacks, or defects. The remarkable result obtained is limited by the high computational needs which introduce additional cost of edge servers.

Furthermore, the authors in [25] proposed a supervised machine learning approach for detection of BSM falsification attack in IoV using the augmented features contained in BSMs collected from trusted neighbor vehicles and the suspicious vehicle at every time instance. The approach achieved high precision and recall but the drawback of this approach is the assumption that some vehicles are trusted in the network.

Finally, the authors in [26] proposed an unsupervised learning approach to detect position falsification attacks using VeReMi dataset to train two deep learning algorithms: gated neural network (GRU) and LSTM. The first model consists of 1-layer GRU and 1- layer LSTM stacked together and the other models consists of a stacked 2-layers LSTM and a stacked of 5 layers LSTM. They evaluated the models based on recall and F1-Score and suggested deploying the proposed model on the edge. The major shortcoming of this approach is the computational requirements.

The comparison table of the reviewed literature in this study is present in Table 1. **Error! Reference source not found.**As it can be observed from the table, the existing literature on data-centric misbehavior detection frameworks in the Intelligent Transportation System (ITS) has primarily focused on utilizing features from a single BSM data for training and prediction. However, this approach leaves room for potential attackers to manipulate and falsify Basic Safety Messages (BSMs) without detection. Therefore, there is a clear need for a new approach that takes into account the limitations of existing frameworks and addresses the challenge of identifying and mitigating misbehavior in a more robust manner.

Table 1 Comparison of Related Works in the Literature with the Proposed Approach

Ref	Dataset Used	ML algorithms Used	Training Speed measured?	Prediction Speed measured?	Precision >99	Recall >99?	MDS does not depend on individual vehicle’s BSM Data
[13]	VeReMi	KNN, DT, LR, SVM, RF	×	×	√	√	×
[15]	VeReMi	NB, KNN, DT, RF	×	×	×	×	×
[27]	VeReMi	LR, DT, RF, NB, KNN, SVM	×	×	×	×	×
[16]	VeReMi	LR, Ensemble, RF, NB, KNN, SVM	×	×	√	×	×
[18]	VeReMi	SVM and LR	×	×	×	×	×
[28]	VeReMi	Deep Learning	×	×	×	×	×

[20]	VeReMi	Ensemble RF	√	√	√	×	×
[21]	BurST-ADMA, VeReMi	Ensemble learning	√	√	√	×	×
[29]	BurST-ADMA	AdaBoost	√	√	√	×	×
[22]	NGSIM	CNN, LR and SVM	×	×	×	×	×
[23]	VeReMi	GTN	×	×	√	√	×
[24]	VeReMi Extension	Deep learning	√	√	√	×	×
[30][26]	VeReMi Extension	Deep Learning	×	×	√	√	×
Proposed Method	BurST-ADMA	DL, DT, RF	√	√	√	√	√

3. THE PROPOSED NEIGHBOR PUBLIC TRANSPORTATION VEHICLE APPROACH

In this section, the IoV network model, the dataset used, data preprocessing, proposed approach and the methodology are discussed.

3.1. IoV System Model

In this paper, we consider a fully functional ITS with licensed public transportation vehicles, such as public busses and trams, having a unique pseudonym pattern different from other vehicles without compromising their privacy. Each vehicle in the network uses its OBU to exchange BSM with other vehicles (V2V). Vehicles also communicate with strategically located RSUs in the network (V2I) and are also capable of communicating with pedestrians via vehicle-to-pedestrian (V2P) and vice-versa via dedicated short-range communication (DSRC) or cellular V2X [31]. The RSUs are also connected to the cloud using a high-speed communication link as shown in Figure 1. The RSUs also maintain a small database for storing previously received BSMs with the vehicle pseudo-ID of all sender vehicles to allow easy query and retrieval of public transportation vehicles BSMs using their special pseudo-ID.

3.1.1. Attack Model

The attacker model in this paper is categorized into four (4) based on how the attacker is involved in the IoV network as follows:

a. Active attackers are the malicious attackers that participate in performing attack in the IoV network by sending bogus information and manipulating some parameters over communication, for example replay attack

b. Passive attackers monitor and collect traffic data such as personally identifiable information (PII) from the network. For example, vehicle tracking

c. An outsider (External) attacker is an attacker who does not possess any cryptographic security credentials for authentication to communicate in the network.

d. An insider (internal) attacker is most dangerous attacker because this attacker possesses all cryptographic security credentials and therefore authenticated to communicate in the network. In the case of IoV, an internal attacker is registered by TA and thus has all the privileges of all other users in the network, making it very challenging to detect this type of attacker. This type of attacker can only be detected using a data-centric detection system such as the one proposed in this paper.

In the intelligent transportation system (ITS) domain, attacks (or misbehavior) can be malicious or benevolent. Malicious attacks are intentional with the aim of causing mayhem, while misbehavior can be as a result of faulty vehicle sensor such as geographic information system (GPS) sensors mounted on vehicles. In this paper, we detect all forms of attacks, whether malicious or misbehavior, due to faulty vehicle sensors.

3.2. Dataset Description

The dataset used in this study is the popular public dataset known as BurST-ADMA[32]. The dataset was generated using the Burwood simulation of urban mobility (SUMO) [33] traffic scenario in the suburb of Burwood in Melbourne, Victoria, Australia over a period of 1000 seconds with BSMs recorded after 1-second interval. Each BSM contains the longitude (x), latitude (y), vehicle’s ID, timestep, speed, acceleration, heading and label. These kinematic features are all important for detecting the attacks in this dataset because all the attacks are related to falsifying position or speed data in the BSM as shown in Table 2. The dataset simulated a traffic scenario consisting of cars, trucks, motorcycles, public buses, public trams and pedestrians. It consists of 207,315 BSMs of which 28,189 are attack. New attacks were added compared to the previous vehicular public dataset for false data injection, making it suitable for IoV scenarios. There are 8 categories of attacks, as can be seen in Table 2**Error! Reference source not found.** and the distribution of each attack category is also presented in that table.

The first attack presented in the dataset is a constant random position attack in which the attacker increments the x and y (latitude and longitude values) of the vehicle by random value. This attack is similar to constant random speed attack with the only difference being incrementing the vehicle speed rather than position. On the other hand, in both positive and negative position offset attacks, the attacker increments or decreases the radius 500m from the original position, while in positive and negative speed offset attacks, the attacker increments or decreases 10m/s from the original speed of the vehicle. Finally, the last attack in this dataset is reversed heading attack in which the attacker adds 180 degrees offset to the original heading in the BSM data to make it look as if a vehicle traveling forward is actually reversing.

3.2.1. Dataset Creation

The three datasets (1-NPTV, 2-NPTV, and 3-NPTV datasets) used in this study were generated from the BurST-ADMA dataset using Algorithm 1**.**

The first dataset generated is the 1-NPTV dataset, which depicts a scenario in which a vehicle is near a single NPTV (a low traffic scenario). This dataset consists of all the features of non-NPTV (simply referred to as vehicle in this paper) BSM, nearest NPTV BSM, and the distance between them. Therefore, 1-NPTV dataset contains the following features: (i-ii) both vehicle and neighbor transport vehicle (NPTV) IDs, (iii-vi) latitude and longitude of vehicle and NPTV, (vii-xii) their speed, heading and acceleration, (xiii) the distance between NPTV and the vehicle, and (xiv) the label of the vehicle.

The second dataset is the 2-NPTV dataset which portrays a medium traffic scenario in which a vehicle is close to two NPTV in traffic. The dataset consists of vehicle BSM features, two nearest NPTV BSM features, and the distance between the vehicle and each of the nearest NPTV. The 2-NPTV dataset comprises of: (i-iii) vehicle and two neighbor transport vehicle (NPTV) IDs, (iv-viii) latitude and longitude of the vehicle and NPTVs, (ix-xv) their speed, heading, and acceleration (xvi-xvii) the distance between the two nearest NPTVs and the vehicle, and (xviii) the label of the vehicle.

Finally, the 3-NPTV dataset is also generated the same way as 1-NPTV and 2-NPTV datasets. This generated dataset illustrates a high traffic scenario in which the vehicle is near three different NPTVs. Therefore, the generated 3-NPTV dataset contain features from vehicle's BSM, three NPTV BSMs, and the distance between the vehicle and the nearest three NPTVs as follows: (i-iv) vehicle and the three neighbor transport vehicles (NPTV) IDs, (v-xii) latitude and longitude of vehicle and the three NPTVs, (xiii-xxiv) their speed, heading and acceleration, (xxv-xxvii) the distance between the three nearest NPTV and the vehicle, and (xxviii) the label of the vehicle.

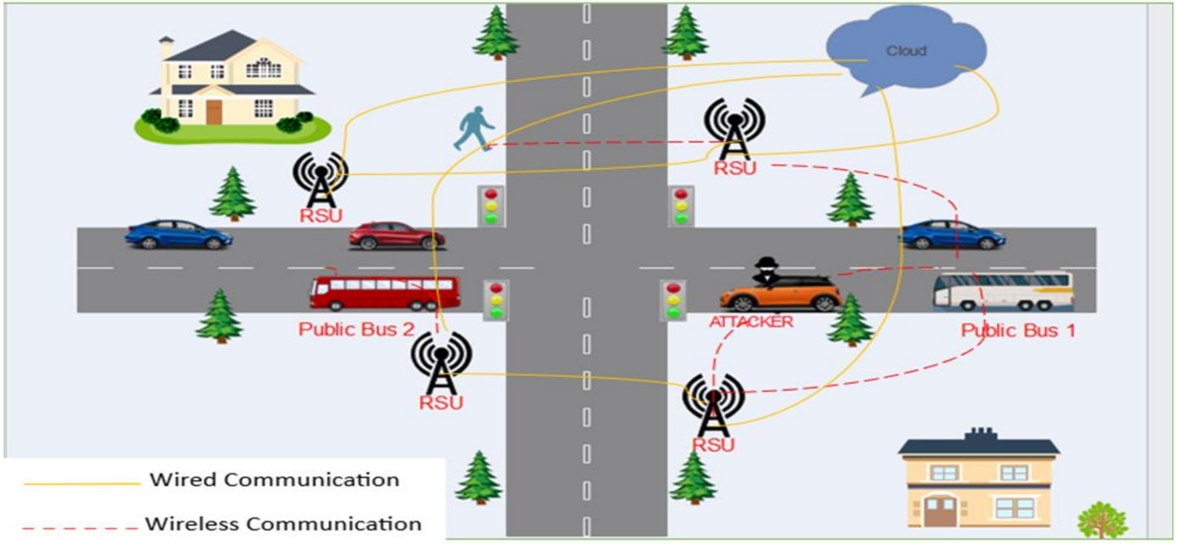

Figure 1 IoV System Model

Table 2 BurST-ADMA Dataset Attack Type Distribution

Label	Type	Number of samples
0	Normal	179126
1	constant random position	2110
2	positive position offset	4130
3	negative position offset	4512
4	constant random speed	4106
5	positive speed offset	4720
6	negative speed offset	4755
7	reversed heading	3856

3.3. Proposed Approach

In the context of the Internet of Vehicles (IoV), vehicles disseminate Basic Safety Messages (BSMs) to exchange important information for safety applications and traffic management in intelligent transportation system. These BSMs contain various features that characterize the vehicle's behavior, position, and other relevant data.

To determine the closest neighboring public transport vehicle (NPTV), the proposed approach utilizes the haversine formula[34] given by equation (1),(2) and (3). This formula calculates the distance between two points on the Earth's surface, considering their latitude and longitude coordinates. By applying the haversine formula, the distance between a sending vehicle and an NPTV can be determined accurately.

Mathematically, let BSM_{veh} represent the BSM disseminated by the sending vehicle (non-NPTV), containing a feature set F = {f_1, f_2, ..., f_n}. Additionally, let BSM_{nptv} represent the BSM disseminated by the closest NPTV, containing a feature set G = {g_1, g_2, ..., g_n}. The proposed approach combines these two BSMs, specifically in the 1-NPTV scenario, to form an augmented feature set H = {F ∪ G}. This union operation merges the feature sets from both BSM_{veh} and BSM_{nptv}, resulting in a comprehensive feature set for training the machine learning classifiers.

In the 2-NPTV scenario, let BSM_{nptv1} and BSM_{nptv2} represent the BSMs disseminated by the two closest NPTVs, with feature sets G1 = {g_1, g_2, ..., g_n} and G2 = {h_1, h_2, ..., h_n}, respectively. The sending vehicle's (non-NPTV) BSM, BSM_{veh}, has the feature set F = {f_1, f_2, ..., f_n}. To create the augmented feature set H for training the machine learning classifiers, we perform the union operation H = {F ∪ G1 ∪ G2}. This combines the features from the sending vehicle and both NPTVs into a single augmented feature set.

Similarly, for the 3-NPTV scenario, let BSM_{nptv1}, BSM_{nptv2}, and BSM_{nptv3} represent the BSMs disseminated by the three

employed to determine the closest NPTV accurately. By merging the feature sets of the sending vehicle and the closest

closest NPTVs, with feature sets G1 = {g_1, g_2, ..., g_n}, G2 = {h_1, h_2, ..., h_n}, and G3 = {i_1, i_2, ..., i_n}, respectively. The sending vehicle's BSM, BSM_{veh}, has the feature F = {f_1, f_2, ..., f_n}. The augmented feature set H for training the machine learning classifiers is obtained by performing the union operation: H = {F ∪ G1 ∪ G2 ∪ G3}.

In all cases, the merging of the feature sets from the non NPTV vehicle and the NPTVs results in an augmented feature set that encompasses the characteristics of all vehicles involved. This expanded feature set provides a more comprehensive representation of the data, enabling the machine learning classifiers to capture a wider range of patterns and relationships.

By merging the feature sets, the proposed approach takes advantage of the cooperative nature of IoV. It prevents a single vehicle's BSM data, which could be manipulated or maliciously tuned by attackers, from solely determining the prediction outcome. Instead, the combined feature set provides a more reliable basis for prediction, incorporating information from both the sending vehicle and the closest NPTV.

The advantages of this approach in an Intelligent Transportation Systems (ITS) environment are numerous. Firstly, it enhances the accuracy of attack detection and classification by utilizing a broader range of features. By considering the behavior of both the sending vehicle and the NPTV, the models can capture more diverse patterns and relationships.

Secondly, the cooperative nature of the approach aligns well with the principles of ITS. It leverages the collective information from multiple vehicles to improve the overall detection capability and prevent an individual vehicle's data from dominating the prediction outcome.

In summary, the proposed approach utilizes BSMs disseminated by vehicles in the IoV. The haversine formula is

NPTV, the approach creates an augmented feature set for training machine learning classifiers. This cooperative approach not only enhances the accuracy of attack detection and classification but also aligns with the principles of ITS, utilizing the collective information from multiple vehicles.

$$a = \sin^2\left(\frac{\varphi B-\varphi A}{2}\right) + \cos\varphi A * \cos\varphi B * \sin^2\left(\frac{\lambda B-\lambda A}{2}\right) \quad (1)$$

$$c = 2 * \arcsin(\sqrt{a}) \quad (2)$$

$$D = R \cdot c \quad (3)$$

Where φA is latitude of BSM received from non-NPTV (BSM_{veh}), φA is the latitude of BSM received from NPTV (BSM_{NPTV}), λA is longitude of BSM_{veh} , λB is longitude of BSM_{NPTV} and R is earth’s radius (mean radius = 6,371km), and D is the distance calculated between BSM_{veh} and BSM_{NPTV}.

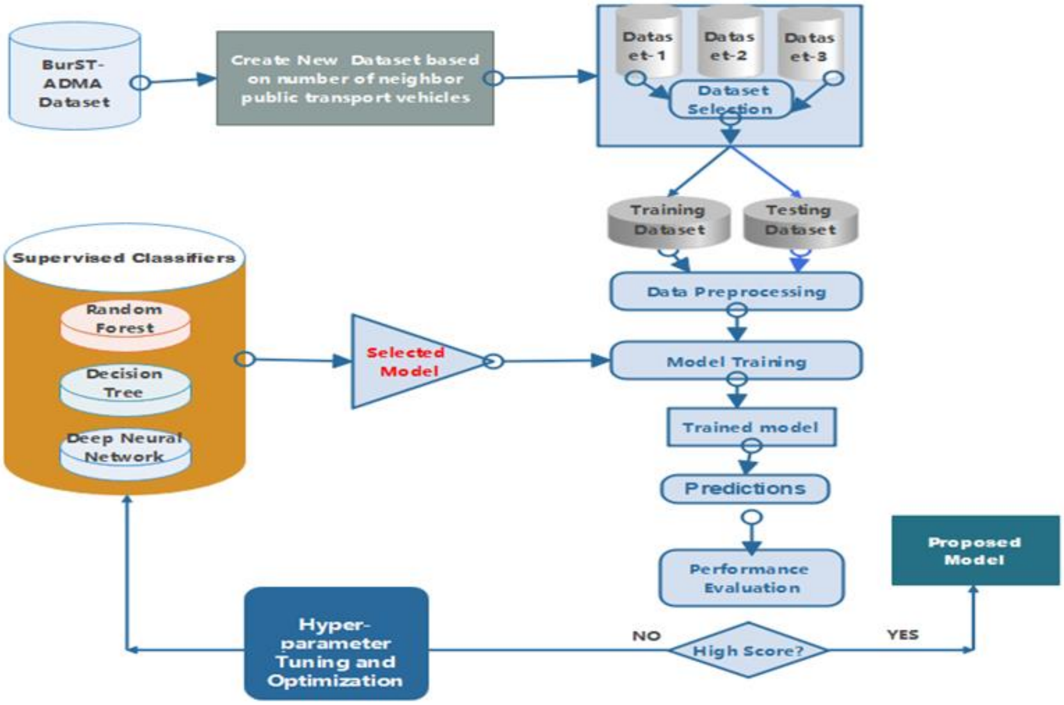

Figure 2 Methodology of the Proposed Approach

In a typical IoV environment, the operation mechanism of the proposed method is described step-by-step in the flowchart shown in Figure 3. In a nutshell, an RSU receives BSM from a random vehicle at time ***t***, $BSM_{Veh,}$ and queries a database, DB-BSM**,** for all the BSMs received from NPTV, BSM_{NPTVi} at the same timeframe (from ***t***-α to ***t***) using NPTV pseudonym (assumed to have a certain pattern for all NPTVs) , where *i* is the number of BSM_{NPTV} received within *t*-α timeframe. The distance D_i between BSM_{VT} and $BSM_{NPTVi.}$ is calculated, sorted and mapped to the corresponding BSM_{NPTV}. The new dataset consists of BSM_{VT}, ***n*** number of BSM_{NPTVi} with their corresponding distances Di according to the traffic scenario, explained in the dataset description section. The augmented feature set obtained is passed on for selection of important features before the features are scaled. Furthermore, the proposed MiDFUPTVA is used to make prediction on the data to determine whether the BSM is received from a benign vehicle or an attacker. If the BSM is predicted to be an attack, the BSM data is forwarded to the misbehavior authority (MA) in a misbehavior report (MR) for appropriate action, such as blacklisting or certificate revocation. This framework is encouraged to be deployed at RSUs for their computational advantage and stricter security.

While the NPTV approach proposed in this study may exhibit similarities to the methodology described by the authors of [25], their approach relies on the assumption of trusted vehicles, which may not align with a zero-trust policy. This assumption introduces vulnerabilities if trust flags are compromised or falsified, potentially leading to unauthorized manipulation within the IoV network. In contrast, NPTV approach does not rely on pre-established trust flags associated with specific types of vehicles. Instead, it utilizes neighbor public transport vehicles with no assumption of pre-established trust. By considering the dynamic interactions between vehicles and their surrounding environment, our approach adopts a more adaptive and context-aware approach to security, mitigating the reliance on static trust assumptions and aligning with the principles of zero-trust security, where every interaction is verified and authenticated, regardless of the source.

In addition, the work in [25] also introduces scalability issues due to the centralized nature of the shared database and trust verification process. Managing and updating trust flags for every registered vehicle can become challenging as the size of the IoV network grows, potentially leading to delays or inefficiencies in trust verification processes. In contrast, our approach adopts a decentralized approach to security and decision-making, leveraging the cooperative nature of the IoV network. By distributing the computation and analysis tasks across roadside units (RSUs), our approach can scale more effectively to accommodate a larger number of vehicles and dynamic traffic conditions.

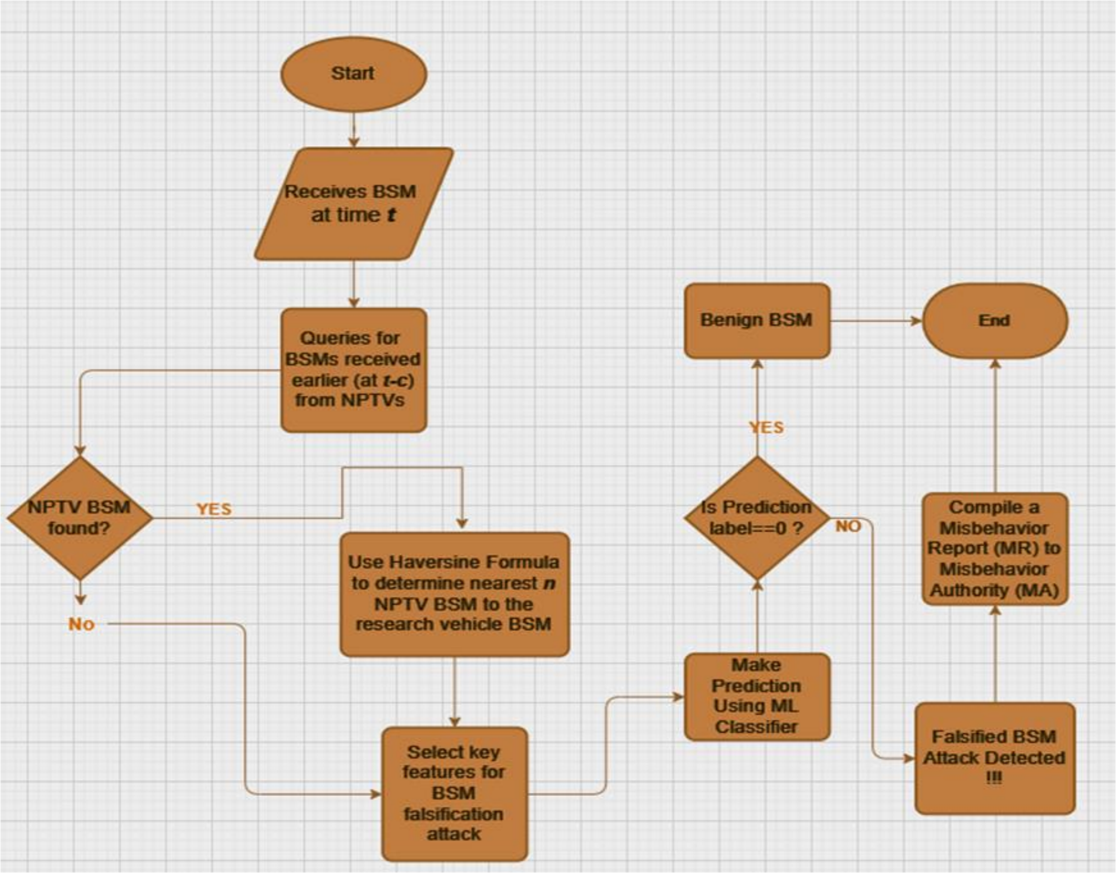
Start
Receives BSM at time t
Queries for BSMs received earlier (at t-c) from NPTVs
NPTV BSM found?
YES
No
Use Haversine Formula to determine nearest n NPTV BSM to the research vehicle BSM
Select key features for BSM falsification attack
Make Prediction Using ML Classifier
Is Prediction label==0 ?
YES
NO
Benign BSM
End
Compile a Misbehavior Report (MR) to Misbehavior Authority (MA)
Falsified BSM Attack Detected !!!

Figure 3 Flowchart for MiDFUPTVA Operation Mechanisms

```
Begin
Input:
$BSM \leftarrow BurST - ADMA\ Dataset$
OUTPUTS:
1-NPTV-dataset, 2-NPTV-dataset, 3-NPTV-dataset
dataset ← list of $BSM_{veh}$, $BSM_{NPTV}$, and the distance between them concatenated together along column axis to form one row of data
STEPS:
$BSM_{NPTV}$ ← BSM [id ==”ptv” | ”p-tram”]
3-NPTV-dataset← nptv_creation(dataset,nptv=3)
$BSM_{Veh}$ ← BSM[~$BSM_{NPTV}$]
for each time in ( BSM[timestep]) do
For each bsm, i in enumerate(BSM[time]) do
If (bsm== $BSM_{veh}$) then
distance=calculate_distance(bsm,$BSM_{NPTVi}$)
dataset[i]=concat(bsm,$BSM_{NPTVi}$,distance, axis=column)
End If
1-NPTV-dataset ← $nptv_{creation}$ (dataset,nptv=1)
2-NPTV-dataset ← nptv_creation (dataset,nptv=2)
End for
End for
```

Algorithm 1 NPTV Dataset Creation Pseudocode from BurST-ADMA

3.4. The Methodology

In this section, the methodology adopted in this paper is discussed in terms of dataset generation, data preprocessing, model selection, training and validation, hyperparameter tuning and model testing.

The first phase of the methodology starts with dataset generation based on *n* NPTVs (where *n=1…3*) using algorithm 1, as shown in Figure 1. The algorithm takes as input the BurST-ADMA labeled dataset described in section 3.2 and searches the ID column for BSMs with ptv (stands for public transportation vehicle) or p-tram (stands for public tram) in the vehicle ID at every time instance in the *timestep* column (these BSMs are termed $BSM_{NPTV)}$. In addition, the algorithm (i) uses the Haversine formula to calculate the distance between a non-NPTV BSM (termed as BSM_{veh}) and all BSM_{NPTV} received at the same timeframe (ii) maps the calculated distances to the corresponding BSM_{NPTV} and (iii) sorts the BSM_{NPTV} from closest to BSM_{veh} to farthest. Therefore, For 1-NPTV dataset, only the BSM_{veh} and closest BSM_{NPTV} with its mapped distance are considered and for 2-NPTV dataset, BSM_{veh} and two nearest BSM_{NPTV} with their mapped distances are considered. Likewise, for the 3-NPTV scenario dataset, the BSM_{veh} and three nearest BSM_{NPTV} with their mapped distances make up the dataset.

The data preprocessing phase takes the labelled datasets generated from the previous phase, checks for missing values, and imputes them with appropriate values based on domain knowledge. Moreover, in this phase, duplicate records are also removed to prevent the model from being biased. Furthermore, synthetic minority oversampling technique (SMOTE) is employed to balance the imbalanced dataset generated. In this phase, feature scaling is performed using scikit-learn's MinMaxScaler according to the equation (4) to ensure that all the features are of the same scale.

$$Xscaled = \frac{Xi-Xmin}{Xmax-Xmin} \quad (4)$$

where XScaled is the scaled feature, Xi is the ith feature, Xmax and Xmin are maximum and minimum values of the ith feature, respectively. The phase concludes with data split into 70% for training and the remaining 30% for testing using the scikit-learn library with a stratify sampling for equal representation of each attack category. The testing data is further split into 70% for training and 30% for validation. The hyper-parameter tuning phase sets up the chosen machine learning classifiers with their corresponding hyper-parameters such as criterion, n_estimators, min_samples_split ,max_depth, number of hidden layers, activation function, number of epochs, and batch-size with possible values and the best score parameters are selected as illustrated in Table 3**.** The hyperparameters are sampled using the scikit-learn's RandomizedSearchCV and validated using K-Fold cross-validation. Random Forest (RF), Decision Tree (DT), and deep learning (DL) models were selected for their excellent performance as observed in the literature.

The training phase uses the hyperparameters with the best score from the previous phase according to selected ML classifier and trains each of the ML classifiers and the DL on the training data with scikit-learn's K-Fold cross-validation employed for validation of the model's performance using the validation data. The learned models are also saved using the pickle library.

Finally, the model testing phase concludes the workflow by loading the learned models to make predictions on the test data. Because accuracy is not a sufficient metric to measure our models due to the imbalanced nature of the ratio of attacks to benign samples in real-world scenarios, the macro-average of precision, recall,F1-Score, and accuracy are used as evaluation metrics. Results visualization is performed using Python's matplotlib and Seaborn libraries.

Table 3 Randomized Search CV Result for Hyper-Parameters used in this Paper

ML Classifier	Hyperparameters Possible Values	Hyperparameters Best Values
	N_estimators={50,100,200}	N_estimators=50
RF	criterion={gini,entropy, log loss }	criterion=gini
	min_samples_split={2,4,8}	min_samples_split=2
	max depth={20,50,100}	max_depth=50
DT	criterion={gini,entropy, log loss }	criterion='log loss'
	Min_samples split={2,4,8}	Min_samples_split=4

	Num_hidden_layers={2,3,4}	Num_hidden_layers=4
	Epochs={20,50,100}	Epochs=50
DL	Batch_size={32,64,128}	Batch_size=64
	Activation_func={tanh,relu}	Activation function=relu

3.5. Experimental Setup

This section explains the experimental setup used in this paper. We employed the use of scikit-learn library for (i) data split into training and testing (ii) Training, validation, hyper-parameter tuning and testing of traditional machine learning algorithms (Random Forest and Decision tree classifiers) and (iii) Evaluation of model performance. We also used Keras from TensorFlow library for training, testing and visualizing accuracy and loss. The machine learning and deep learning models were trained on Dell Precision 5520 with Processor Intel(R) Core (TM) i7-7820HQ CPU @ 2.90GHz, 2901 Mhz, 4 Core(s), 8 Logical Processor(s) and 16GB of RAM. All the algorithms are trained as multiclass classifiers (label 0-7) as shown in Figure 2. Each classifier is trained and tested using four scenarios namely zero-Neighbor public transport vehicle (zero-NPTV), i.e. the raw dataset, one-Neighbor public transport vehicle (1-NPTV), two-Neighbor public transport vehicle (2-NPTV) and three-Neighbor Public Transport Vehicle (3-NPTV) to investigate different scenarios a vehicle can find itself in real world traffic and examine the effects on detection rate which demonstrates the adaptability and flexibility of the proposed framework. The training and prediction time per samples are calculated and the performance of the three models (multiclass) are evaluated using macro average of accuracy, precision, recall and f1-score according to the following equations (5 -8):

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

$$F1 - Score = 2 \times \frac{Pre \times Recall}{Pre+Recall} \quad (8)$$

Where TP, TN, FP, and FN are true positive, true negative, false positive and false negative respectively.

4. RESULT ANALYSIS AND DISCUSSION

This section presents the results obtained from the experiment carried out employing the proposed misbehavior detection framework using neighbor public transport vehicle approach (MiDFUPTVA). The experiment was conducted using the Python programming language's scikit-learn library and TensorFlow. The result analysis and visualization were performed using Python's matplotlib and Seaborn library. The selection of RF and DT as the chosen machine learning algorithms was based on their frequent usage in the literature and their superior performance compared to other algorithms employed in the dataset used by the authors in [32].

4.1. Zero-Neighbor Public Transport Vehicle

In this section, we conducted experiments using the BurST-ADMA dataset to train Deep Learning (DL), Random Forest (RF), and Decision Tree (DT) classifiers. However, it is important to note that these classifiers were trained solely on individual Basic Safety Message (BSM) data without incorporating the neighbor public transport (NPTV) approach proposed in this study.

Among the three classifiers, the RF classifier achieved the highest accuracy score of 97.86%. However, its precision was only 96.38%, and the recall was even lower at 91.36%. This indicates that in a real-world Internet of Vehicles (IoV) scenario, the RF classifier would likely exhibit a high rate of false negatives, potentially compromising the effectiveness of misbehavior detection.

The DT classifier, on the other hand, achieved an accuracy score of 96.52%, with slightly over 92% precision and recall. While these metrics are relatively better compared to the RF classifier, they still fall short in terms of performance.

The DL classifier, which was trained on the raw dataset without applying the NPTV approach, exhibited relatively low performance across all metrics. It achieved approximately 92% accuracy, precision, recall, and F1-Score. This can be attributed to the limited number of features available in the raw dataset before the incorporation of the NPTV approach. Overall, the performance scores obtained by the classifiers, as depicted in Figure 5, strongly indicate the necessity of the proposed NPTV approach. The results underscore the importance of leveraging the NPTV approach to enhance the performance and effectiveness of misbehavior detection in the IoV environment.

4.2. 1-Neighbor Public Transport Vehicle (1-NPTV)

In this particular scenario, the proximity of the sending vehicle to a single neighboring public transport vehicle (1-NPTV) was considered. The performance of three classifiers, namely Random Forest (RF), Decision Tree (DT), and Deep Learning (DL), was evaluated in this context. The experimental setup section of this paper provided insights into the architecture of the deep learning model used throughout the study.

During the training and testing phases, the classifiers were trained on a dataset consisting of 145,120 samples and tested on 62,195 observations. The results revealed that both DT and RF classifiers outperformed the DL model in terms of accuracy, precision, recall, and F1-score. DT achieved scores of 98.64%, 95.55%, 95.45%, and 95.47% in these metrics, while RF achieved scores of 99.36%, 99.76%, 99.46%, and 97.99% respectively, as presented in Table 3. On the other hand, the DL model attained accuracy, precision, recall, and F1-score scores of 92.62%, 94.30%, 53.75%, and 59.85% respectively. The DL model's relatively poor recall performance in misclassifying attack type 3 (negative position offset) and attack type 7 (reversed heading) contributed to its lower overall performance. The limited number of features available in the dataset might have hindered the DL model's ability to effectively learn the patterns of the attacks, highlighting the need for our proposed approach to incorporate more features in the dataset and improve the DL model's performance.

It should be noted that while RF demonstrated the best overall performance, DT exhibited the highest training and prediction speed, making it a favorable choice for embedded environments like vehicles' On-Board Units (OBU). This observation is further supported by the data presented in Table 4. Additionally, a comparison of the three models across all metrics is depicted in Figure 4, providing a visual representation of their respective performance.

4.3. 2-Neighbor Public Transport Vehicle (2-NPTV)

In this particular scenario, we considered a situation where a sending vehicle is surrounded by two neighboring public transport vehicles simultaneously (2-NPTV). The evaluation of the classifiers in this context revealed interesting findings. The DL model, which was enhanced with the inclusion of additional features to learn attack patterns, demonstrated significant improvements in performance. This can be observed in Table 3, where the DL model exhibited excellent scores across various metrics. However, it is worth noting that this enhancement came at a cost, as the DL model experienced a drawback in terms of training speed, measured by the number of training samples per second, as well as prediction speed. Both training and prediction speeds were more than twice as slow compared to the 1-NPTV scenario, as indicated in the same table.

On the other hand, the DT model achieved nearly perfect scores in all metrics, indicating its exceptional performance. Additionally, the DT model boasted the highest training and prediction speeds among the three classifiers. However, it is important to mention that the DT model recorded a recall score of 98.8%, primarily due to misclassifying attack type 6 (negative speed offset).

Meanwhile, the RF model delivered outstanding results, surpassing the 99% threshold in all metrics. Its performance was commendable across the board. A visual comparison of the three models in terms of accuracy, precision, recall, and F1-Score is presented in Figure 5, providing a comprehensive understanding of their respective performances.

These findings highlight the trade-offs associated with each classifier and shed light on their strengths and limitations in the 2-NPTV scenario. The DL model showcased improved performance with the incorporation of additional features but suffered from slower training and prediction speeds. The DT model exhibited exceptional accuracy and speed, but experienced challenges in correctly classifying attack type 6. On the other hand, the RF model excelled in all metrics, demonstrating its robustness and reliability in this scenario.

4.4. 3-Neighbor Public Transport Vehicle (3-NPTV)

In this scenario, the machine learning (ML) algorithms were trained using Basic Safety Messages (BSMs) collected from the sending vehicle as well as three neighboring public transport vehicles (3-NPTV). The performance of the DL, RF, and DT models was evaluated in this context, yielding interesting results. Notably, both the DL and RF models outperformed the DT model, which can be attributed to the availability of sufficient features in the dataset to effectively learn the patterns of attacks, including the more sophisticated attack type 7. A detailed analysis of the performance metrics can be found in Table 3.

Compared to the previous scenarios, it is important to highlight that all models showcased slower training and prediction speeds in this scenario. This can be attributed to the increased number of features included in the dataset, as presented in Table 4. Despite the slower speeds, the DL and RF models demonstrated exceptional performance, surpassing the DT model in terms of accuracy, precision, recall, and F1-score. The availability of a more comprehensive dataset with a richer feature set contributed to the improved performance of these models.

Interestingly, although the DT model obtained a lower overall score, it exhibited the best prediction and training speeds among the three models. This suggests that the DT model may be a favorable choice in scenarios where speed is a critical factor, such as embedded environments like vehicles' On-Board Units (OBU).

A visual comparison of the three models' performance in this scenario is presented in Figure 6, providing a clear illustration of their relative strengths. Additionally, Table 4 provides detailed information on the prediction and training speeds achieved by the models throughout the study. It is worth noting that the dataset sizes varied across the different scenarios, with 145,120 samples used for training and 62,195 samples for testing in the 1-NPTV scenario, 16,624 samples for training and 1,851 samples for testing in the 2-NPTV scenario, and 29,210 samples for training and 12,530 samples for testing in the 3-NPTV scenario. The training and testing speeds were measured in terms of the number of observations processed per second, providing insights into the computational efficiency of the models in each scenario.

4.5. Discussion

In this research, we employed Random Forest (RF), Decision Tree (DT), and Deep Learning (DL) models, keeping them unchanged throughout the study. The primary objective was to showcase the impact of our proposed data-centric approach. Unlike the model-centric approach, which focuses on tweaking model parameters to improve performance, our approach centers around enhancing the data itself to achieve better results.

Our data-centric approach involved incorporating features from multiple Basic Safety Message (BSM) data in different IoV scenarios, namely 1-NPTV, 2-NPTV, and 3-NPTV. By utilizing BSM data from one or more neighboring public transport vehicle (NPTV), we aimed to capture a broader range of information and increase the model's ability to detect and classify attacks accurately.

The results of our study consistently demonstrated that incorporating features from more than one BSM data yielded superior performance compared to the zero-NPTV scenario. This finding underscores the power of our proposed approach. By leveraging the information from multiple NPTVs, we provided the models with a more comprehensive and diverse set of features, enabling them to learn and recognize attack patterns more effectively.

Our data-centric approach offers several advantages. Firstly, it allows us to leverage the collective knowledge and information from multiple sources, enhancing the overall understanding of the data. This holistic perspective enables the models to capture a broader range of attack patterns, including more complex and sophisticated attacks.

Secondly, the inclusion of features from NPTVs provides a more robust and reliable foundation for the models' training and prediction processes. The incorporation of diverse data helps mitigate biases and limitations that may be present in individual data from one BSM, leading to improved generalization and performance. It is important to note that our approach does not rely on fine-tuning the model parameters but rather focuses on enriching the data itself. This distinction highlights the significance of the data-centric perspective in addressing the challenges associated with attack detection and classification. In summary, our data-centric approach, which incorporates features from NPTVs, has proven to be powerful in enhancing the performance of the RF, DT, & DL models (Table 5). By expanding the scope and depth of the data, we provide the models with a richer understanding of the attack patterns, resulting in improved accuracy and effectiveness. This approach offers valuable insights into the field of attack detection and classification, highlighting the importance of considering the data itself as a critical factor in achieving superior performance.

Table 4 Average Performance of the Classifiers in Relation to Different Number of NPTVs

Traffic Scenario	ML Classifier	Accuracy	Precision	Recall	F1-Score
zero-NPTV	RF	0.97.86	0.9638	0.9136	0.9436
	DT	0.9652	0.9218	0.9203	0.9210
	DL	0.9185	0.9228	0.9236	0.9232
1-NPTV	RF	0.9936	0.9976	0.9636	0.9799
	DT	0.9864	0.9551	0.9545	0.9547
	DL	0.9262	0.9430	0.5375	0.5985
2-NPTV	RF	0.9995	0.9997	0.9941	0.9968
	DT	0.9989	0.9997	0.9881	0.9936
	DL	0.9897	0.9474	0.9615	0.9537
3-NPTV	RF	0.9992	0.9965	0.9861	0.9909
	DT	0.9681	0.9933	0.9757	0.9831
	DL	0.9998	0.9996	0.9995	0.9961

Table 5 Comparison between DL, DT and RF for Training and Prediction Speed

Traffic Scenario	ML Classifier	Training Speed (Obs/s)	Prediction Speed (obs/s)
zero-NPTV	RF	2605.85	035009.85
	DT	32684.68	144639.54
	DL	64.74	1150.1
1-NPTV	RF	4520.87	16365.10
	DT	290821.64	698820.22
	DL	735.16	18846.97
2-NPTV	RF	8749.473	4883.91
	DT	99544.91	31372.88
	DL	557.85	8731.13
3-NPTV	RF	3489.85	3952.68
	DT	27819.05	21237.29
	DL	379.70	6265

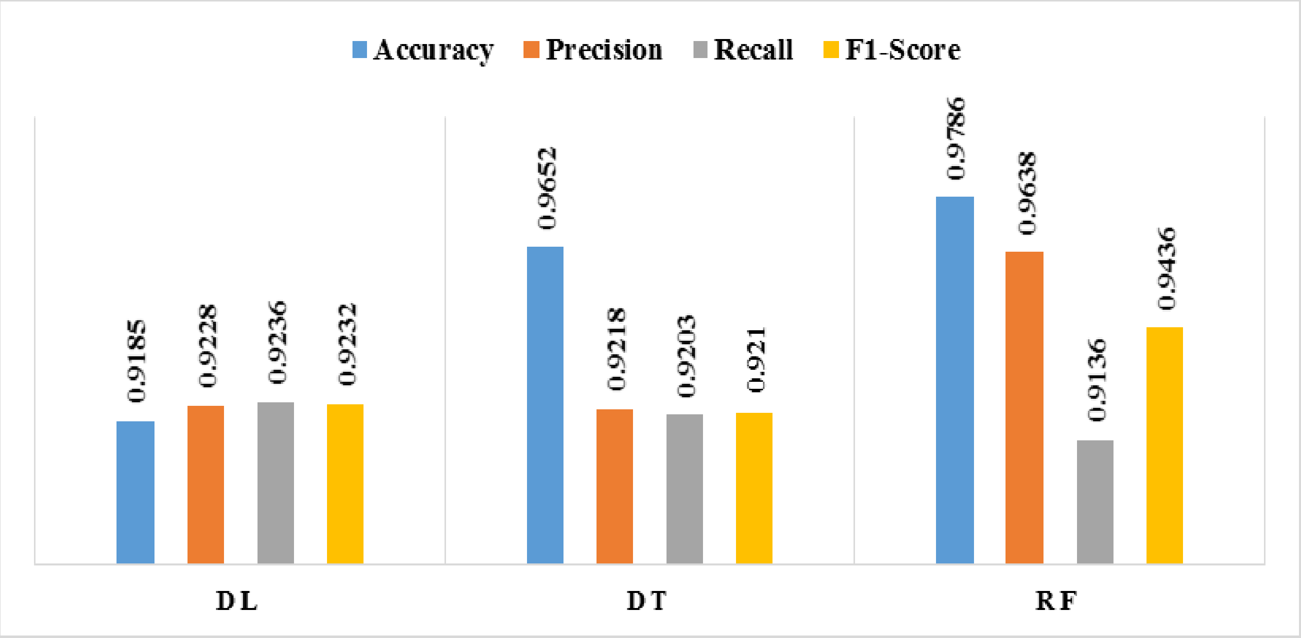

Figure 4 Average Performance Comparison for DL, DT and RF in zero-NPTV Scenario

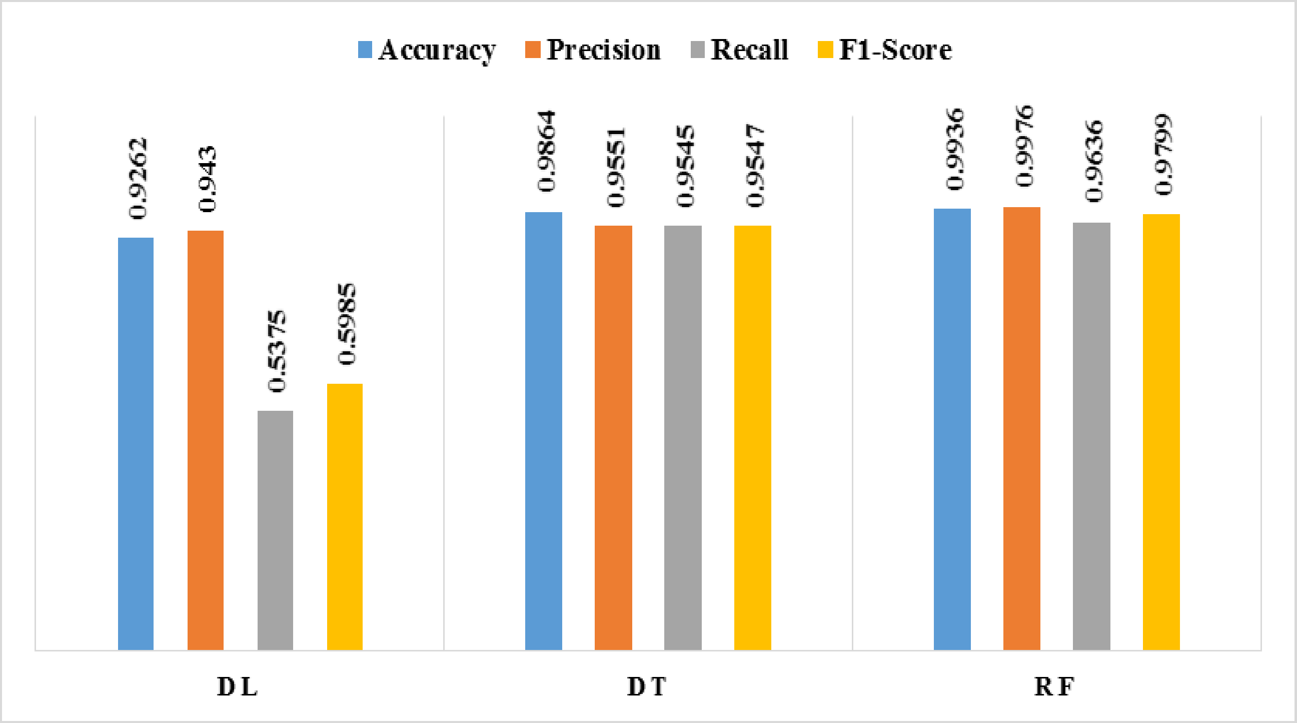

Figure 5 Comparison of Average Performance for DL, DT and RF in 1-NPTV Scenario

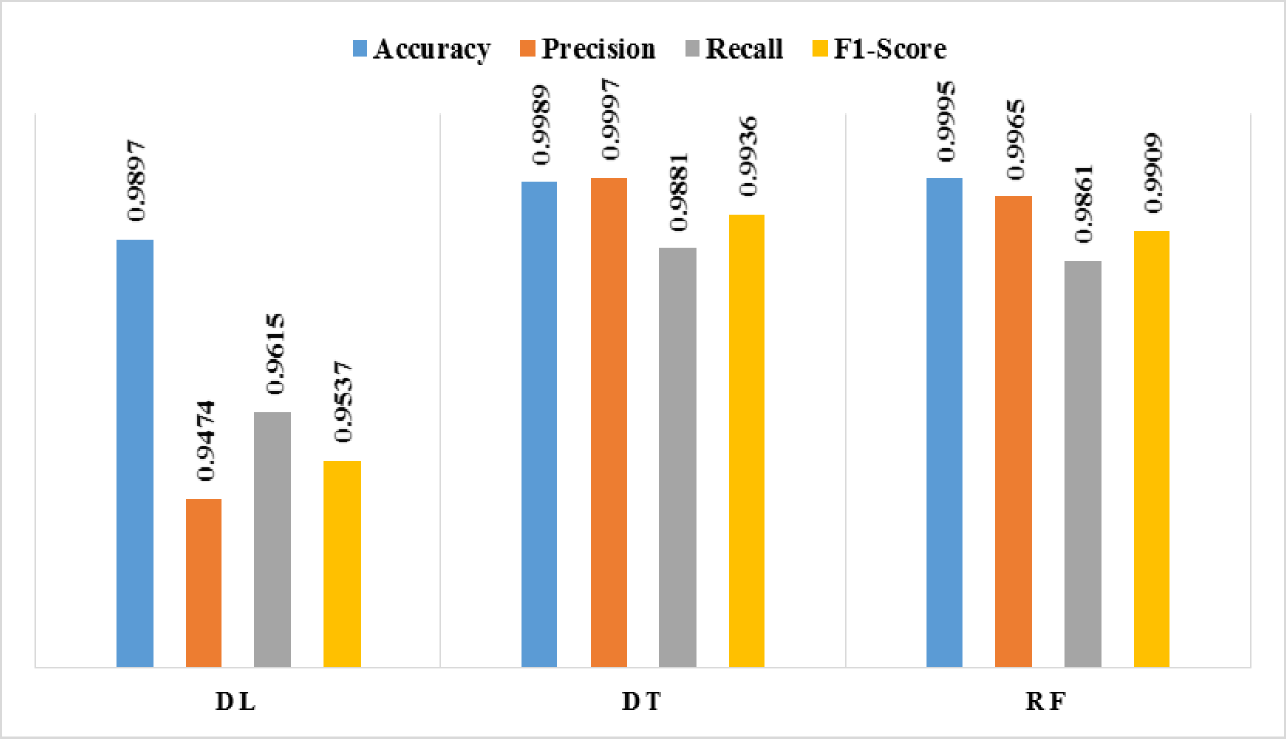

Figure 6 Comparison of Average Performance for DL, DT and RF in 2-NPTV Scenario

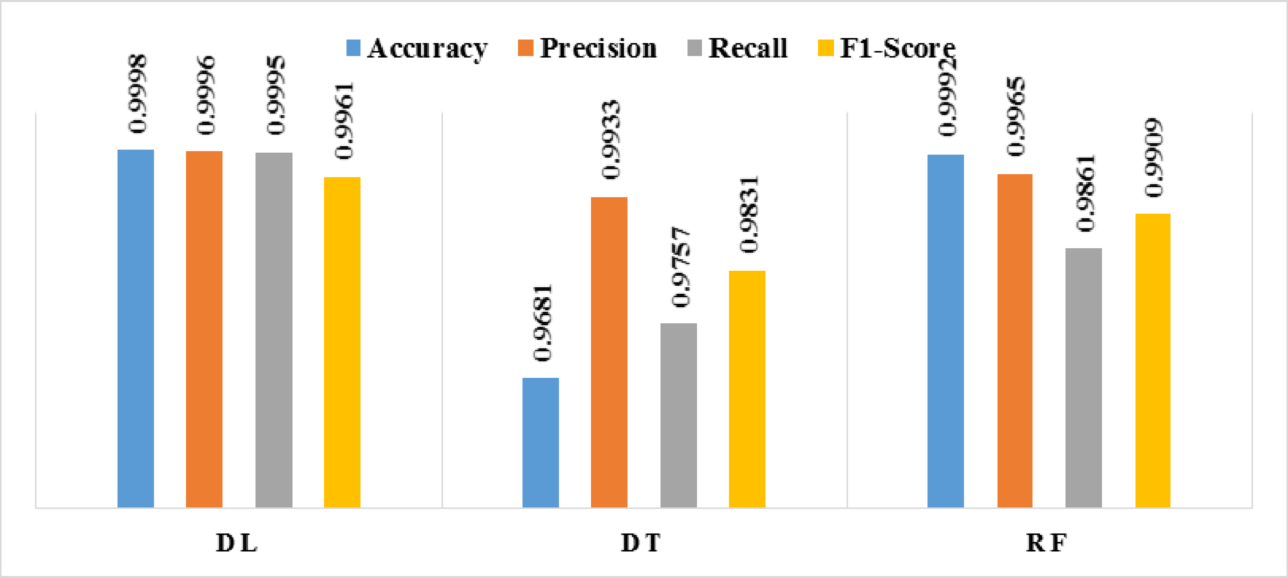

Figure 7 Comparison of Average Performance for DL, DT and RF in 3-NPTV Scenario

4.6. Comparison with State-of-the-art Techniques

To justify our proposed work, we compared our proposed misbehavior detection framework using the public transport vehicle approach (MiDFUPTVA) with related works in the literature that used the same BurST-ADMA public dataset in their proposed frameworks for the detection of false data injection attacks in IoV. We found that our proposed approach obtained higher scores in all the metrics, as shown in Table 6. To the best of our knowledge, the chosen state-of-the-art works are the most recently published works that used the same dataset to evaluate their work; hence, we chose these works for justification. We first compared our approach against the raw BurST-ADMA dataset [32], in which DT and RF obtained the best results among other ML classifiers and only their work measured all the metrics. Furthermore, we compared our work against the optimized AdaBoost work in [21] , and finally ,we compare the results presented by the authors of the work presented in [29].

Table 6 Comparison between our Work and Prior Work

Ref	ML Used	Accuracy	Precision	Recall	F1-Score
BurST-ADMA [32]	RF	99.63%	99.88%	97.75%	98.75%
	DT	99.01%	96.63%	96.75%	96.75%
[21]	RSO-FDS	99.90%	NA	NA	NA
[29]	AdaBoost	98.9%	NA	NA	NA
Proposed Method	DT(2-NPTV)	99.89%	99.97%	98.81%	99.36%
	RF(3-NPTV)	99.92%	99.65%	98.61%	99.09%
	DL(3-NPTV)	99.98%	99.96%	99.95%	99.61%

5. CONCLUSION AND FUTURE WORK

The emergence of IoV and its integration into cooperative intelligent transportation system has improved ITS in safety and non-safety applications. In this paper, we proposed a novel data-centric misbehavior detection framework using a public transport vehicle approach for the detection of false data injection attacks. We successfully displayed different scenarios in which a sending vehicle may find itself in a traffic to demonstrate the flexibility and scalability of the proposed framework. We used deep learning, decision tree, and random forest classifiers and obtained a near- perfect score using our approach. We also observed that the reason for the almost perfect score in our work and the related work found in the literature that used the same dataset is due to the low noise and few features in the dataset (all the features have high correlation with the attacks). However, our approach used all the important features related to position and speed (since all the attacks are related to speed and position data falsification) and augmented the features with NPTV's position and speed data for a more robust model and to mitigate the effect of low noise and few features in the dataset. In the future, we plan to explore more attacks in IoV, especially attacks on safety applications such as adaptive cruise control. We would also work to analyze how using licensed public transport vehicles or high occupancy vehicles (HOV) can enhance security and optimize traffic and reduce carbon emissions by reducing the number of vehicles on the roads, thereby pushing us toward green transportation.

REFERENCES

[1] L. Ecola, S. W. Popper, R. Silberglitt, and L. Fraade-Blanar, "Road to Zero: Developing A Vision for a Future with Zero Roadway Fatalities," 2019.

[2] S. A. Elsagheer Mohamed and K. A. AlShalfan, "Intelligent Traffic Management System Based on the Internet of Vehicles (IoV)," J. Adv. Transp., vol. 2021, p. 4037533, 2021, doi: 10.1155/2021/4037533.

[3] S. A. Elsagheer Mohamed et al., "Safe Driving Distance and Speed for Collision Avoidance in Connected Vehicles," Sensors, vol. 22, no. 18, 2022, doi: 10.3390/s22187051.

[4] S. Chen et al., "Vehicle-to-Everything (v2x) Services Supported by LTE-Based Systems and 5G," IEEE Commun. Stand. Mag., vol. 1, no. 2, pp. 70–76, 2017, doi: 10.1109/MCOMSTD.2017.1700015.

[5] I. S. Committee and others, "IEEE standard for wireless access in vehicular environments (WAVE)-networking services," IEEE Std, vol. 1609, pp. 3–2010, 2007.

[6] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. Ben Jemaa, and P. Urien, "Simulation Framework for Misbehavior Detection in Vehicular Networks," IEEE Trans. Veh. Technol., vol. 69, no. 6, pp. 6631–6643, Jun. 2020, doi: 10.1109/TVT.2020.2984878.

[7] B. Brecht and T. Hehn, "A Security Credential Management System for V2X Communications," in Connected Vehicles: Intelligent Transportation Systems, R. Miucic, Ed. Cham: Springer International Publishing, 2019, pp. 83–115. doi: 10.1007/978-3-319-94785-3_4.

[8] A. Boualouache and T. Engel, "A Survey on Machine Learning-based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks," IEEE Commun. Surv. Tutorials, p. 1, 2023, doi: 10.1109/COMST.2023.3236448.

[9] T. Garg et al., A Survey on Machine Learning-based Misbehavior Detection Systems for 5G and Beyond Vehicular Networks, vol. 3, no. 5. 2022, pp. 1–14. doi: 10.1109/OJVT.2021.3138354.

[10] S. A. E. Mohamed, "Automatic traffic violation recording and reporting system to limit traffic accidents: based on vehicular Ad-hoc networks (VANET)," in 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), 2019, pp. 254–259.

[11] S. A. Elsagheer, Y. Atallah, and H. Hashem, "Enhancing Road Safety Using the Internet of Vehicles: A Machine Learning-Based Collision Detection Approach," in 2023 11th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC), 2023, pp. 68–72.

[12] S. Sharma and B. Kaushik, "A survey on internet of vehicles: Applications, security issues & solutions," Veh. Commun., vol. 20, p. 100182, 2019, doi: https://doi.org/10.1016/j.vehcom.2019.100182.

[13] A. Sharma and A. Jaekel, "Machine Learning Based Misbehaviour Detection in VANET Using Consecutive BSM Approach," IEEE Open Journal of Vehicular Technology, vol. 3. pp. 1–14, 2022. doi: 10.1109/OJVT.2021.3138354.

[14] J. Kamel, M. Wolf, R. W. Van Der Hei, A. Kaiser, P. Urien, and F. Kargl, VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs, vol. 2020-June. 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149132.

[15] A. Sharma and A. Jaekel, "Machine Learning Approach for Detecting Location Spoofing in VANET," in 2021 International Conference on Computer Communications and Networks (ICCCN), 2021, vol. 2021-July, pp. 1–6. doi: 10.1109/ICCCN52240.2021.9522170.

[16] P. Sharma and H. Liu, "A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles," IEEE Internet Things J., vol. 8, no. 6, pp. 4991–4999, Mar. 2021, doi: 10.1109/JIOT.2020.3035035.

[17] S. So, P. Sharma, and J. Petit, "Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET," in 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 564–571. doi: 10.1109/ICMLA.2018.00091.

[18] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect position falsification attack in vanets," in Security and Privacy: Second ISEA International Conference, ISEA-ISAP 2018, Jaipur, India, January, 9--11, 2019, Revised Selected Papers 2, 2019, pp. 166–178.

[19] H. Grover, T. Alladi, V. Chamola, D. Singh, and K.-K. R. Choo, "Edge Computing and Deep Learning Enabled Secure Multitier Network for Internet of Vehicles," IEEE Internet Things J., vol. 8, no. 19, pp. 14787–14796, Oct. 2021, doi: 10.1109/JIOT.2021.3071362.

[20] G. O. Anyanwu, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Novel hyper-tuned ensemble Random Forest algorithm for the detection of false basic safety messages in Internet of Vehicles," ICT Express, 2022, doi: https://doi.org/10.1016/j.icte.2022.06.003.

[21] G. O. Anyanwu, C. I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Falsification Detection System for IoV Using Randomized Search Optimization Ensemble Algorithm," IEEE Trans. Intell. Transp. Syst., pp. 1–15, 2023, doi: 10.1109/TITS.2022.3233536.

[22] S. A. Almalki and F. T. Sheldon, "Deep Learning to Improve False Data Injection Attack Detection in Cooperative Intelligent Transportation Systems," in 2021 IEEE 12th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2021, pp. 1016–1021. doi: 10.1109/IEMCON53756.2021.9623153.

[23] Y. Wu, L. Wu, and H. Cai, "A deep learning approach to secure vehicle to road side unit communications in intelligent transportation system," Comput. Electr. Eng., vol. 105, p. 108542, 2023, doi: https://doi.org/10.1016/j.compeleceng.2022.108542.

[24] T. Alladi, B. Gera, A. Agrawal, V. Chamola, and F. R. Yu, "DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs," IEEE Trans. Veh. Technol., vol. 70, no. 11, pp. 12013–12023, 2021, doi: 10.1109/TVT.2021.3113807.

[25] H. A. Idris, K. Ueda, B. Mokhtar, and S. A. E. Mohamed, "Novel Intelligent BSM Falsification Attack Detection System Using Trusted Neighbor Vehicle Approach in IoV," Int. J. Comput., vol. 23, no. 1, pp. 116–125, Apr. 2024, doi: 10.47839/ijc.23.1.3443.

[26] T. Alladi, V. Kohli, V. Chamola, and F. R. Yu, "A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems," Digit. Commun. Networks, 2022, doi: https://doi.org/10.1016/j.dcan.2022.06.018.

[27] F. Hawlader, A. Boualouache, S. Faye, and T. Engel, "Intelligent Misbehavior Detection System for Detecting False Position Attacks in Vehicular Networks," in 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Jun. 2021, pp. 1–6. doi: 10.1109/ICCWorkshops50388.2021.9473606.

[28] X. Wang, Y. Zhu, S. Han, L. Yang, H. Gu, and F.-Y. Wang, "Fast and Progressive Misbehavior Detection in Internet of Vehicles Based on Broad Learning and Incremental Learning Systems," IEEE Internet Things J., vol. 9, no. 6, pp. 4788–4798, Mar. 2022, doi: 10.1109/JIOT.2021.3109276.

[29] G. O. Anyanwu, C. I. Nwakanma, J.-H. Kim, J.-M. Lee, and D.-S. Kim, "Misbehavior Detection in Connected Vehicles using BurST-ADMA Dataset," in 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Oct. 2022, pp. 874–878. doi: 10.1109/ICTC55196.2022.9952947.

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

[31] Y. L. Morgan, "Notes on DSRC \& WAVE standards suite: Its architecture, design, and characteristics," IEEE Commun. Surv. \& Tutorials, vol. 12, no. 4, pp. 504–518, 2010.

[32] M. A. Amanullah, M. Baruwal Chhetri, S. W. Loke, and R. Doss, "BurST-ADMA: Towards an Australian Dataset for Misbehaviour Detection in the Internet of Vehicles," 2022, pp. 624–629. doi: 10.1109/PerComWorkshops53856.2022.9767505.

[33] C. Sommer et al., "Veins: The open source vehicular network simulation framework," in EAI/Springer Innovations in Communication and Computing, A. Virdis and M. Kirsche, Eds. Springer, 2019, pp. 215–252. doi: 10.1007/978-3-030-12842-5_6.

[34] C. C. Robusto, "The cosine-haversine formula," Am. Math. Mon., vol. 64, no. 1, pp. 38–40, 1957.

Authors

Hussaini Aliyu Idris a final year MSc research student at the Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology Alexandria, Egypt. He obtained his bachelor degree in Computer Engineering from Arab Academy for Science, Technology and Maritime Transport Alexandria, Egypt. His research interests include machine learning, intelligent transportation system (ITS), Smart City, internet of vehicles, AI for sustainability, Big Data and data science.

Kazunori Ueda received the M. Eng. and Dr. Eng. degrees in information engineering from the University of Tokyo in 1980 and 1986, respectively. He joined Waseda University, Department of Information and Computer Science in 1993, and has been Professor since 1997 till now and he is a visiting Professor at Egypt-Japan University of Science and Technology (E-JUST) since 2010. His current research interests include design and implementation of programming languages, concurrency and parallelism, high-performance verification, and hybrid systems. His recent project is LMNtal (pronounced "elemental"), a model of concurrency and a concurrent programming language based on hierarchical graph rewriting, whose implementation has now evolved into a model checker with visualizing tools.

Bassem Mokhtar received his PhD in computer engineering from Virgina Tech USA in 2014 and is currently an Assistant professor at United Arab Emirate University. He published several papers in the different research areas including but not limited to network, Vehicular ad-hoc networks (VANETS), intelligent transportation and smart city.

Samir A. Elsagheer Mohamed received his PhD in computer Networks from Université de Rennes I in 2013 and is currently an Associate professor at the Department of Computer Science and Engineering, Egypt-Japan University of Science and Technology Alexandria, Egypt. He published several papers in the different research areas including but not limited to network, Vehicular ad-hoc networks (VANETS), intelligent transportation, cyber-security, internet of vehicles and smart city.

How to cite this article:

Hussaini Aliyu Idris, Kazunori Ueda, Bassem Mokhtar, Samir A. Elsagheer Mohamed, "Machine Learning Based Misbehavior Detection System for False Data Injection Attack in Internet of Vehicles Using Neighbor Public Transport Vehicle Approach", International Journal of Computer Networks and Applications (IJCNA), 11(2), PP: 159-176, 2024, DOI: 10.22247/ijcna/2024/224442.