Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems

Javed Ashraf, Asim D. Bakhshi, Nour Moustafa, *Senior Member, IEEE*,
Hasnat Khurshid, Abdullah Javed, and Amin Beheshti

***Abstract*— Intelligent Transportation Systems (ITS), especially Autonomous Vehicles (AVs), are vulnerable to security and safety issues that threaten the lives of the people. Unlike manual vehicles, the security of communications and computing components of AVs can be compromised using advanced hacking techniques, thus barring AVs from the effective use in our routine lives. Once manual vehicles are connected to the Internet, called the Internet of Vehicles (IoVs), it would be exploited by cyber-attacks, like denial of service, sniffing, distributed denial of service, spoofing and replay attacks. In this article, we present a deep learning-based Intrusion Detection System (IDS) for ITS, in particular, to discover suspicious network activity of In-Vehicles Networks (IVN), vehicles to vehicles (V2V) communications and vehicles to infrastructure (V2I) networks. A Deep Learning architecture-based Long-Short Term Memory (LSTM) autoencoder algorithm is designed to recognize intrusive events from the central network gateways of AVs. The proposed IDS is evaluated using two benchmark datasets, i.e., the car hacking dataset for in-vehicle communications and the UNSW-NB15 dataset for external network communications. The experimental results demonstrated that our proposed system achieved over a 99% accuracy for detecting all types of attacks on the car hacking dataset and a 98% accuracy on the UNSW-NB15 dataset, outperforming other eight intrusion detection techniques.**

***Index Terms*— Intelligent transport systems, CAN bus, internet of vehicles, autonomous vehicles, intrusion detection system, deep learning, LSTM, autoencoder.**

I. Introduction

INTELLIGENT Transport Systems (ITS) aim at making effective use of communication and information technologies to accomplish a considerable decrease in pedestrians and traffic accidents and improve the transport management system. Autonomous vehicles (AVs) and connected vehicles, which form major components of ITS, would result in minimizing the requirements of drivers, lessen the traffic fatalities and transportation expenditures and enhancing traffic flow. AVs can communicate with other vehicles, where AVs are linked to communication technologies that are commonly termed as Vehicle to Everything (V2X). V2X can take the form of Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Pedestrian (V2P), and Vehicle-to-Network (V2N) technologies [1]. These technologies enable a quicker flow of important transportation data related to traffic, drivers and vehicles, which not only help in making transportation systems more convenient but also promote innovations in the use of these technologies. With the availability of multiple types of wireless communication means, V2X is capable of connecting to numerous other internet of things (IoT) devices [2].

When AVs are connected to the internet (collectively called as the Internet of Vehicles (IoVs)), the IoT devices and their communication technologies bring inherent security vulnerabilities [3], [4] related to involved IoT devices and their communication technologies. AVs are vulnerable to network attacks which may have serious consequences as controlling the vehicles moving on roads, maliciously, will have serious risks to human lives. Security threats to IoVs can be broadly divided into two categories: 1) Attacks related to networks connecting the vehicles to the external world, and 2) In-Vehicles Network(IVN) Attacks [5], [6] [7].

The most common attacks to IoVs include Denial of Service (DoS), Distributed DoS (DDoS), replay, spoofing, and Brute force. DoS and DDoS attacks can be launched by flooding the nodes with superfluous messages resulting in overloading of the nodes, which then make the nodes unavailable to process legitimate requests. Replay attacks are executed by intercepting the traffic to identify the unencrypted data which facilitates hackers to perform Man-in- the- Middle (MITM) attack. Another type of attack that can be targeted in IoVs is a spoofing attack, where attackers can masquerade as valid users and forward the fake GPS data to the nodes. Furthermore, the spoofing attack can attempt to interrupt the communication between two nodes, resulting in attacks similar to DoS attacks [8]. Brute force attack is another type of attack that can be launched to crack passwords in IoV networks. Also,

Manuscript received May 21, 2020; revised July 25, 2020; accepted August 13, 2020. This work was supported by Dr. Nour Moustafa's Fellowship, the 2020 Australian Spitfire Memorial Defence Fund, under Grant PS39150. The Associate Editor for this article was A. Jolfaei. *(Corresponding author: Nour Moustafa.)*

Javed Ashraf, Asim D. Bakhshi, and Hasnat Khurshid are with the Department of Computer Software Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan (e-mail: javed.ashraf@mcs.edu.pk; asim.dilawar@mcs.edu.pk; hasnat@mcs.edu.pk).

Nour Moustafa is with the School of Engineering and Information Technology, University of New South Wales at ADFA, Canberra, ACT 2612, Australia (e-mail: nour.moustafa@unsw.edu.au).

Abdullah Javed is with the Sir Syed Centre for Advanced Studies in Engineering (CASE), Institute of Technology, Islamabad 44000, Pakistan (e-mail: abdullahjaved.case@gmail.com).

Amin Beheshti is with the Department of Computing, Macquarie University, Sydney, NSW 2109, Australia (e-mail: amin.beheshti@mq.edu.au).

Digital Object Identifier 10.1109/TITS.2020.3017882

different types of attacks can be launched against IoV networks and systems in the routing process, like DoS, eavesdropping, masquerading, and route modification [9].

In addition to external network attacks mentioned above, IoVs are also vulnerable to attacks launched from intra-vehicle and in-vehicle networks (IVN). For in-vehicles communications, Controller Area Network (CAN) [10] is the defacto standard used for communication between vehicles' electronic control units (ECUs). A CAN bus protocol enables connecting ECUs with actuators and sensors to simplify the implementation of multiple applications of AVs and IoVs. Apart from providing an error detection mechanism for smooth communications, control and diagnostic information shared through CAN bus facilitates automotive services, like Advanced Driver Assistance Systems (ADAS) and autonomous driving [11]. However, since CAN bus communications do not involve any authentication mechanism, attackers can access the bus, inject malicious messages and take control of the bus to launch further attacks.

The nodes, unavoidably, can handle the messages without authenticating their source. Likewise, infotainment service, which is available in almost all modern cars, is vulnerable to be exploited by attackers to launch malicious activities through the over-the-air (OTA) update module. This could result in malfunctions of vehicles and may prove to be fatal for the driver, other vehicles as well as for the pedestrians. Fuzzy attacks can be launched on IVNs by introducing random messages forcing the vehicles to malfunction or demonstrate unexpected or undesired behavior [12]. Similar to attacks launched on IoVs from external networks, spoofing and DoS attacks can also be launched on IVNs through flooding the network with fake messages as well as forwarding erroneous information related to, for example, RPM or gear which can result in fatal accidents and loss of human lives. In order to protect IoVs against threats to IVNs from physical access or attacks through external communications, there is a requirement to design a robust detection system which can detect suspicious network events generated from an external network as well as by ECUs.

Intrusion detection system (IDS) offers a robust monitoring capability that recognizes anomalous behaviors and messages sharing through the communication of devices in external networks and between vehicles [13], [14]. Machine Learning (ML) and Deep Learning (DL) techniques have been extensively used to develop IDSs for conventional networks and partly in IoVs networks and systems [15]. However, ML or DL techniques face major challenges, in particular when handling the heterogeneity of IoVs and IoT devices [16]–[18], which can be connected to IoVs. Also, such systems need a huge dataset covering attack and normal observations to develop binary or multi-classification IDSs which can classify existing attack events with good accuracy. Most of the available IDSs fail to detect new or zero-day attacks as their signatures are not held at the time of training the models [19]. Thus, an anomaly-based IDS is the most suitable option for the protection of IoVs which can recognize existing as well as new attacks since it builds a profile from the normal data and any deviation from that profile is detected as an anomaly. However, designing such a model that can capture all possible patterns or behaviors of normal data is highly challenging. This is because such models tend to bias towards the dominated class, i.e., the normal class, producing high false positive rates (FPR) [20], [21]. At the same time, it is also impossible to capture all possible patterns of normal observations that may occur in IoVs systems and networks, which then increases false negative rates [22] [23].

DL algorithms, in particular, new variants of Artificial Neural Networks (ANNs), have become a popular solution for designing IDSs [24], [25]. This is because ANNs are capable of learning complex patterns that make them suitable for distinguishing between normal and attack traffic on the network. However, apart from being heavy on computational resources, ANNs are often trained in a supervised manner, thus not sufficiently suitable to detect unknown attacks. With the above challenges, we propose a long short-term memory (LSTM) based auto-encoding IDS which can learn to detect known as well as unknown attacks from IoVs networks. LSTM network architectures are a specialized form of more general recurrent neural networks (RNNs) [26]. The main attribute of LSTM based RNNs is to persist information or cell state for later use in the network. This property makes them suitable for performing analysis of temporal data that changes over time. Thus, LSTM networks are preferred to solve problems related to text translation, image recognition, speech recognition, and also for anomaly detection in time-series sequence data [27].

In this study, there are three main contributions. Firstly, we present a novel statistical feature extraction technique for capturing contextual features from network traffic which can contribute in designing an efficient IDS for IoVs. Secondly, a novel LSTM autoencoder based scheme is proposed to design an IDS for AVs and IoVs. Thirdly, an integrated strategy to detect attacks in both CAN Bus networks as well as in external networks is also suggested. To the best of our knowledge, there is no published work available which proposes non-signature based IDS to detect attacks in both CAN bus as well as from external networks. The only published work which handles the detection of attacks in CAN bus as well as from external networks is proposed by [28], however, the proposed IDS is signature-based and thus can not identify most of the zero-day attacks. Whereas our proposed IDS is non-signature based and aimed at detecting existing as well as new attacks. We hypothesize that the LSTM based auto-encoding can efficiently learn a compressed representation of statistically transformed high-dimensional network flow sequence data.

There are primarily two motivations behind the design of the current architecture. Firstly, the training can be done in an unsupervised manner without *a priori* knowledge of spatio-temporal distribution, statistical characterization or ground-truth of the network flow time-series. Secondly, LSTM autoencoders are well utilized for their robust learning of latent nonlinear representations and subsequent reconstruction [29], [30]. Unlike conventional neural learning approaches, autoencoders are not restricted to dense layers and with LSTM units augment them to work in generative mode with variable length input sequences. The proposed architecture further exploits the fact that univariate sequence representations are best modeled by LSTM autoencoders.

The proposed IDS works in three stages. In the preliminary stage, statistical features of recent network traffic are captured from both the CAN bus as well as external networks. The aim is to capture the underlying parameters of the stochastic complexity of network flow data and use those parameters to detect deviations in the network. The reduced feature space is then fed to the recurrent architecture with hidden LSTM layers as sliding temporal windows in the second stage. After completion of training cycles, training and validation losses are minimized to zero and convergence of weights, the compressed representation of normal traffic is assumed to be sufficiently learned. The learning target of network architecture is to reconstruct input sequences or features that can contribute to the classification as per an optimally set threshold. The reconstruction error is then compared with the threshold to identify if the sequence is anomalous or otherwise. The performance validation of the proposed system is carried out for both anomaly detection in IVNs as well as on an external communication network. To this end, the proposed system is evaluated using two benchmark datasets, i.e., the car hacking dataset [10] and the UNSW dataset [31].

The rest of the paper is organized as follows. Section II describes the related work in the domain of anomaly detection related to AVs and IoVs. In-Vehicle network interfaces and their vulnerabilities are discussed in Section III. In Section IV, the detail of the proposed IDS methodology is explained. In in Section V, results and performance analysis of proposed IDS are described. Finally, the study is summarized in Section VI.

II. Related Work

DL-based IDSs for protection of AVs, connected vehicles and IoVs include new variants of ANN-based models, including convolutional neural networks (CNN), as well as more advanced methods such as generative adversarial networks (GAN), deep belief networks (DBN) and RNN. A recently proposed IDS based on GAN (GIDS) was first trained by supervised learning using labeled dataset for normal and artificially generated noisy data similar to the CAN traffic [10]. GIDS achieved an average accuracy of 98% against each of the four attacks, i.e., DoS, Fuzzy, RPM, and gear attacks. In [32], an LSTM network-based model was applied and evaluated on a CAN traffic data generated from a real vehicle. The proposed system predicted the next likely message from each sender on the bus and most unexpected bits in the actual next messages were flagged as anomalies. It was demonstrated that the system can detect anomalies, with a low false rate.

In [11] a DBN architecture was used to design a classifier evaluated on a simulated dataset.The captured eigenvectors from in-vehicle network data were trained on DNN parameters using both the pre-training method of DBN and the conventional stochastic gradient descent method. Using the probability computed by the DNN for each class, the system detected the attack traffic launched at the vehicle. Another scheme in [33] tried to solve the problem of anomaly detection in CAN bus considering the temporal nature of the network traffic data. The study employed a hybrid approach by first training a hidden Markov model (HMM) to learn the normal behaviors of a vehicle and then used a regression model to adjust the likelihood threshold for predicting anomalies. The study attempted to identify sophisticated anomalies, which are usually missed out by other methods used for monitoring of CAN Bus, but it consumed high computational resources.

Another hybrid IDS (D2H-IDS) employing DBNs was proposed for connected vehicles by [34]. DBNs were used for data dimensionality reduction and decision trees for classifications of attacks. In [28] the authors proposed an IDS based on a tree-based ML algorithm that is capable of detecting attacks on CAN bus as well as from external networks, but it produced high false alarm rates. Authors also used proposed SMOTE oversampling technique and tree-based averaging approaches for feature reduction and to reduce the computational requirements. However, being dependent on the training of attack signatures, the proposed IDS could not be successful to detect new attacks. A deep convolutional neural network (DCNN) architecture based IDS was proposed to protect CAN bus from spoofing and DoS attacks but it achieved high false-negative and error rates [35]. In recent work published in [36], the authors propose what they call a data-driven IDS in which they used DL architecture based on the CNN to extract features on link loads and detect the intrusion attempts launched at Road Side Unit (RSUs) with the overall aim of protecting IoVs. They tested and evaluated their proposed model and compared the results with NN, SVM and PCA methods on the same dataset and demonstrated that their CNN based proposed model achieved better accuracy as compared to the other three methods.

In [37], the authors present an Intrusion Prevention System (IPS) for IoVs. Their solution is based on Fuzzy Logic and Q-Learning algorithms to detect and prevent DDoS attacks, launched through UDP flooding, in IoVs communication. They place their proposed IPS on 6LoWPAN Border Router (6BR) in the RPL based IoV network by monitoring all types of traffic coming through the IoV network. Although results are not clearly mentioned (in terms of figures and accuracy percentages) in comparison to the existing work, authors claimed to have achieved better accuracy as compared to existing related solutions.

In [38], the authors propose DL based IoV authentication and security monitoring mechanism. In that, first they present fog based IoV authentication method which manages authentication of new vehicles which wants to join the fog. Secondly, they propose Random Forest based DL algorithm for two-way authentication and security monitoring of IoVs. Authors compare their results with ANN and SVM based classifiers and demonstrate that their proposed scheme produced better accuracy for authentication and improved adaptability to a high-speed IoV network environment.

Comparing to existing work related to intrusion detection in IoVs, performance of our proposed IDS is advantageous as follows: 1) It is non-signature based IDS which is capable of detecting attacks in both CAN bus as well as external networks. 2) It can detect zero day attacks. 3) It produced

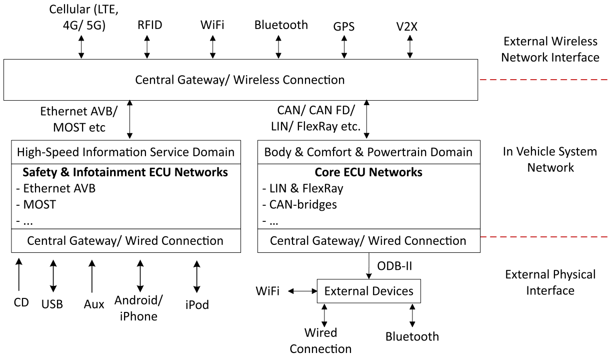

Cellular (LTE, 4G/ 5G)
RFID
WiFi
Bluetooth
GPS
V2X
External Wireless Network Interface
Central Gateway/ Wireless Connection
Ethernet AVB/ MOST etc
CAN/ CAN FD/ LIN/ FlexRay etc.
High-Speed Information Service Domain
Safety & Infotainment ECU Networks
- Ethernet AVB
- MOST
- ...
Body & Comfort & Powertrain Domain
Core ECU Networks
- LIN & FlexRay
- CAN-bridges
- ...
In Vehicle System Network
Central Gateway/ Wired Connection
Central Gateway/ Wired Connection
CD
USB
Aux
Android/ iPhone
iPod
ODB-II
WiFi
External Devices
Wired Connection
Bluetooth
External Physical Interface

Fig. 1. Illustration of IVNs interface types.

better accuracy in terms of detection rate as compared to other ML/DL based IDSs.

III. In-Vehicle Networks and Their Vulnerabilities

Figure 1 illustrates the detailed layout of in-vehicles system connectivity as well as its connectivity to the external world. The in-vehicle electronic system comprises multiple ECUs interconnected through several means of communications, which include IVNs like CAN bridge, local interconnect network (LIN), media-oriented system transport (MOSP) and FlexRay. The networks communicate with external ones through central gateways. In particular, gateways enable intra-vehicle network communication which facilitates a wide range of services [39]. Some of these services include gas and toll payment, vehicle diagnostics, fleet management, automatic collision notification, entertainment/ infotainment and safety services,etc. Also, the on-board diagnostics 2 (OBD2) interface is used to exchange information and status among ECUs. Furthermore, V2I and V2V communication allow improving the safety of cars, drivers, passengers and pedestrians.

As shown in Figure 1, there are various types of external interfaces of the IVN, which includes IVN, external physical network interface and external wireless network interface. Hackers with malicious intent can launch multiple types of attacks on IoVs through IVNs because of their weak security measures, for instance, the absence of authentication and encryption mechanisms, as well as large and an increasing number of interfaces [40]. The attacks can be launched using any of the interfaces, that is, a physical, wired or wireless interface.

The generalized scenarios of attacking AVs or IoVs are depicted in Figure 2. In the first case, an attacker can use OBD-II port to access the CAN bus physically using, for example, a laptop or Raspberry Pi device. Thereafter, this CAN bus compromise can be exploited as an entry point to take control of the target vehicle. In the second scenario, a hacker can attempt to exploit the telematics services of the vehicles provided by the OEM to take control of a telematics device of the attacked vehicle [41]. Once the attacker takes control of any of the CAN nodes, it can control and manipulate the working of critical components of the vehicle, for example, a gear and brake system, killing the engine, jamming the door locks, by injecting manipulated messages, etc.

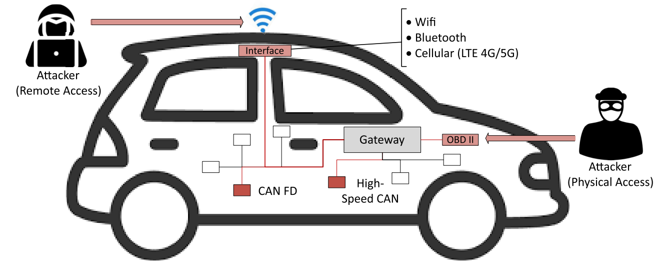


Fig. 2. Description of two hacking scenarios on CAN bus; an attack launched by physical access through OBD-II port and an attack launched by a remote access through one of the external communications means.

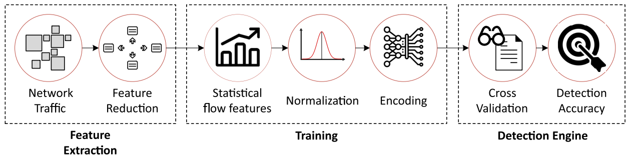


Fig. 3. High-level architecture of proposed IDS for AVs.

To protect against threats of CAN bus, the proposed IDS can be placed on the CAN bus itself to monitor every message for recognizing any suspicious behavior. Alternatively, in order to protect CAN bus against attacks launched through external networks, the proposed IDS can be placed inside the central gateway. Thus, every message coming in and going out of CAN bus is first checked by the IDS at the gateway for discovering any malicious behavior.

Placing an IDS or similar software in vehicles used to be a challenge because of the constrained memory and processing resources of the vehicles. However, due to rapid drop in cost of computing technologies and devices, most modern vehicles, both autonomous and non-autonomous, are equipped with sufficient computing and memory resources allowing deployment of firmware and software, like IDS.

IV. Proposed IDS Architecture

A. System Architecture

The proposed IDS architecture for detecting intrusive events launched against IVNs is shown in Figure 3. It consists of three main stages, i.e., feature extraction, training and validation, and decision engine. The feature extraction stage is responsible for taking raw data as input, extracting network flow features, applying feature reduction methodology, and then extracting statistical data from the reduced features to improve the performance of the proposed deep learning model. The output of this stage is multi-dimensional feature vectors. The statistical features are subsequently fed to the encoding LSTM deep learning layer as sliding temporal windows in the next stage.

The LSTM layer captures the latent temporal and spatial non-linearities in the sequence of feature vectors. The autoencoder layer encodes and reconstructs normal traffic data. At this stage, the reconstruction error, which is the

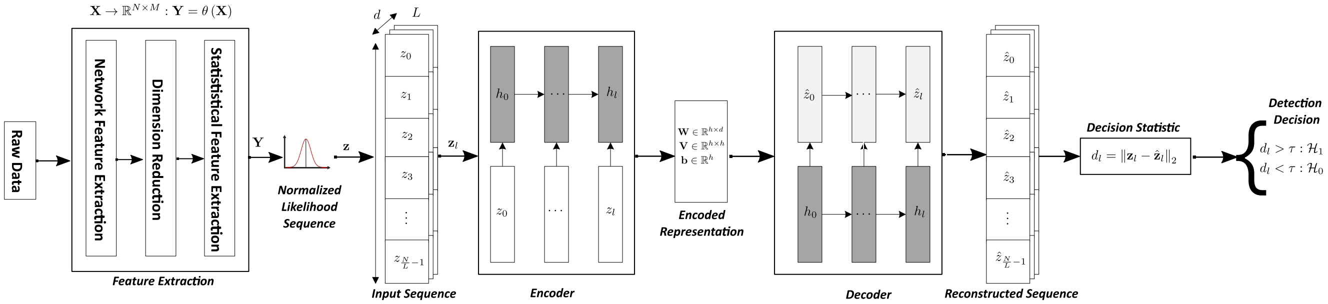

$\mathbf{X} \to \mathbb{R}^{N\times M} : \mathbf{Y} = \theta(\mathbf{X})$
Raw Data
Network Feature Extraction
Dimension Reduction
Statististical Feature Extraction
Feature Extraction
$\mathbf{Y}$
Normalized Likelihood Sequence
$\mathbf{z}$
d
L
z_0
z_1
z_2
z_3
$\vdots$
$z_{\frac{N}{L}-1}$
Input Sequence
$\mathbf{z}_l$
h_0
$\cdots$
h_l
z_0
$\cdots$
z_l
Encoder
$\mathbf{W} \in \mathbb{R}^{h\times d}$
$\mathbf{V} \in \mathbb{R}^{h\times h}$
$\mathbf{b} \in \mathbb{R}^{h}$
Encoded Representation
$\hat{z}_0$
$\cdots$
$\hat{z}_l$
h_0
$\cdots$
h_l
Decoder
$\hat{z}_0$
$\hat{z}_1$
$\hat{z}_2$
$\hat{z}_3$
$\vdots$
$\hat{z}_{\frac{N}{L}-1}$
Reconstructed Sequence
Decision Statistic
$d_l = \|\mathbf{z}_l - \hat{\mathbf{z}}_l\|_2$
Detection Decision
$d_l > \tau : \mathcal{H}_1$
$d_l < \tau : \mathcal{H}_0$

Fig. 4. The workflow of proposed IDS for AVs and their networks; where high-dimensional raw data are filtered by feature extraction, dimensionality reduction and normalized likelihood sequence transformation stages. The outputs of the last stage.

difference between the input and reconstructed sequences, is minimized after multiple training iterations and an optimal threshold is set to classify the sequences based on respective reconstruction errors. In the last stage, the reconstruction errors are computed over a separate validation dataset to determine how far they are from normal sequences. The sequences with reconstruction errors higher than the chosen threshold are labeled as malicious events.

The workflow of the proposed IDS is presented in Figure 4 for discovering anomalies from IoVs and their networks. The raw network flow data is collected from [10] and [31] as pcap and csv files, respectively. The features of the two datasets are extracted which are then used to compute statistical features. The statistical features then input to recurrent networks with hidden LSTM layers as a sliding temporal window. After multiple training iterations, training and validation losses are minimized to zero and the compressed representation of normal traffic is assumed to be sufficiently learned. At this stage, the largest Root Mean Square Error (RMSE) of the reconstruction function is employed as the threshold of classifying normal and attack events. In the detection phase, the reconstruction error is compared with the estimated threshold to identify if the sequence is anomalous or otherwise.

B. Data Preprocessing and Feature Extraction

It is an essential stage to pre-process raw data of IoVs and their network traffic to design an effective intrusion detection methodology. The raw data is filtered to remove any redundancy and missing values. This also includes the extraction of important data features that can be utilized to apply the proposed deep learning-enabled detection method. There are two datasets employed in this study (explained in Section V-B) to validate the credibility of our proposed methodology. The car hacking dataset [10] for detecting anomalies from in-vehicles communications and the UNSW-NB15 dataset [31] for recognizing cyber-attacks from external network traffic.

1) Feature Extraction Stage: The feature extraction stage starts with the selection of data features used for further analysis. The CAN bus dataset is an authentic dataset for validating IDSs of in-vehicles communications. The main attributes of the data field frames are used in the learning model of the IDS. With regards to designing an IDS for protection of IVNs from external networks that are vulnerable to multifarious types of attacks, there is a requirement to consider significant data features. This, however, demands to solve the problem of the curse of dimensionality, that reduces the performance of the IDSs, in terms of processing time and detection accuracy. In the context of IDS to be deployed on CAN bus or central gateway, the computational complexity is addressed by carefully adopting the most important attributes from the external network data, i.e., the UNSW-NB15 dataset, which can model the normal behaviour efficiently so that any deviations can be flagged as cyber-attacks.

In the case of CAN bus messages, there is no destination ID mentioned in the messages. Therefore, two network features are captured to train and validate the intrusion detection model. The first feature is the total number of packets sent by the unique CAN ID in the given time window (3 seconds). The second feature is the size of outbound messages from unique ID. The same features are extracted after every three-second time window from both datasets. These features are then fed to the statistical feature extraction stage to extract the most representative features that contain patterns of attack events.In order to extract the behavior of the hosts which communicated through data packets, a three-second snapshot consisting of additional features is captured to ensure the high performance of the proposed IDS. These features are summarized as follows:

- Traffic initiating from this packet's source IP.
- Traffic sent between this packet's source and destination IPs.
- Time to live from this packet's source to destination and destination to source
- Bits per second for this packet's source to destination and destination to source
- Packets count sent from this packet's source to destination and destination to source

2) Statistical Feature Extraction Stage: Iterating over a three second time window on the data, the number of packets generated is employed to extract more statistical features that enhance the detection accuracy of the proposed deep learning-enabled IDS. At this stage, statistical features from the two datasets are generated. Table I and Table II describe the statistical features of the car hacking dataset and NNSW-NB15 dataset, respectively. The statistical features include mean and standard deviation computed from packets or messages data-frame size and packets' count as shown.

TABLE I

DESCRIPTION OF STATISTICAL FEATURES EXTRACTED FROM CAR HACKING DATASET

Message's Feature	Statistical Feature	Aggregated by	Description
Count	c	CAN ID	Packet count of outbound traffic
Size	μ, σ	Data [0] to Data [7]	Bandwidth of the outbound traffic

TABLE II

DESCRIPTION OF THE STATISTICAL FEATURES EXTRACTED FROM UNSW-NB15 DATASET

Message's Feature	Statistical Feature	Aggregated by	Description
Size	μ, σ	Source_IP	Bandwidth of the outbound traffic
Time Duration	μ, σ	Source_IP	Duration of network flow from unique IP
Time Duration	μ, σ	Source_IP-Destination_IP	Duration of ttl from unique pair of Source-destination IP
Size	μ, σ	Source_IP-Destination_IP	Bandwidth of out-bound and inbound traffic
Count	μ, σ	Source_IP-Destination_IP	Packet count of outbound and inbound traffic

It is hypothesized that statistical characteristics of the normal behavior will not change abruptly and the anomalous or attack behavior will result in a sharp change in statistical features, which can be used to detect an attack in comparison to the threshold of normal data.

C. Training and Detection

1) Normalized Likelihood Transformation: After feature extraction, the network flow can be modeled as an independent and identically distributed $N-$dimensional discrete stochastic process$\{\mathbf{x}_m \in \mathbb{R}^N, m = 1, 2, \cdots, M\}$, given M features and N time instants. Assuming the feature subspace $\mathbf{X} = [\mathbf{x}_0, \cdots, \mathbf{x}_{M-1}] \in \mathbb{R}^{N \times M}$ as an M component univariate mixture, a feature likelihood of each element x_{mn} of $\mathbf{X}$ can be estimated using the probability density function, as given by:

$$f_{\mathbf{x}_m}\left(x_{mn}|\mu_m, \sigma_m^2\right) = \left(2\pi\sigma_m^2\right)^{-\frac{1}{2}} \exp\left(-\frac{(x_{mn}-\mu_m)^2}{\sigma_m^2}\right) \quad (1)$$

where

$$\hat{\mu}_m = \frac{1}{N}\sum_{n=0}^{N-1} x_{mn} \quad (2)$$

and

$$\hat{\sigma}_m = \frac{1}{N}\sum_{n=0}^{N-1}\left(x_{mn} - \hat{\mu}_m\right)^2 \quad (3)$$

where, $\hat{\sigma}_m$ is the moment estimate for the parameters μ_m, σ_m^2 of the univariate Gaussian distribution for mth feature. The resultant likelihood space

$$\mathbf{Y} = \theta(\mathbf{X}) \quad (4)$$

such that, $\mathbf{Y}$ is a parameterized one-to-one transformation obtained through repeated application of (1) on elements of $\mathbf{X}$ as described in Algorithm 1. The new parameterized space $\mathbf{Y} = [\mathbf{y}_0, \cdots, \mathbf{y}_{M-1}] \in \mathbb{R}^{N \times M}$ represents the instantaneous Gaussian likelihood corresponding to each parameter of the data. The aggregate normalized likelihood characterization can then be obtained by:

$$\mathbf{z} = \frac{\sum_{m=0}^{M-1}\mathbf{y}_m}{\max\sum_{m=0}^{M-1}\mathbf{y}_m} \quad (5)$$

where $\mathbf{z} = [z_0, \cdots, z_{N-1}] \in [0, 1]$.

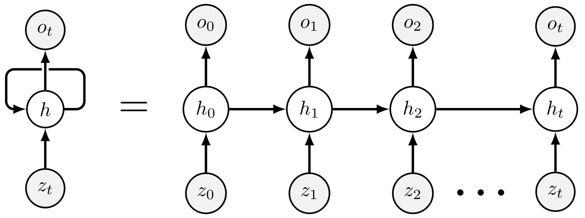


Fig. 5. Structure of a basic recurrent neural network in unfolded form with just one hidden layer.

Algorithm 1 $LikelihoodTransformation(\mathbf{X})$

```
for each col_m X ($m = 0, \cdots, M-1$)
    $\hat{\mu}_m \leftarrow$ compute (2)
    $\hat{\sigma}_m \leftarrow$ compute (3)
    for each col_n $\mathbf{X}^\top$ ($n = 0, \cdots, N-1$)
        $\mathbf{Y} \leftarrow$ compute (1)
    end for
Compute $\mathbf{z}$ using (5)
```

2) Development of LSTM Autoencoder-Based IDS: The transformed normalized likelihood sequence $\mathbf{z}$ can be interpreted as a quasi-time series which captures the temporal dynamics of network flow as well as spatial characteristics of initially extracted features. We propose two different modes of encoding and subsequent reconstruction of these Spatio-temporal characteristics of network flow. In the classic autoencoder model, $\mathbf{z}$ is fed as an input to a fully symmetric connected recurrent neural network [42], which is a basic unfolded structure shown in Figure 5.

This architecture with one hidden layer can be understood as an implementation of the classic multi-layer perceptron working in the auto-associative mode [43]. In this traditional mode, both input, and output layers aim to achieve the joint transition as given by:

$$\begin{aligned} \phi &: \mathbf{z} \to \mathcal{W} := \mathbf{h} = \sigma(\mathbf{W}\mathbf{z} + \mathbf{b}) \\ \psi &: \mathcal{W} \to \hat{\mathbf{z}} = \sigma'\left(\mathbf{W}'\mathbf{h} + \mathbf{b}'\right) \end{aligned} \quad (6)$$

where $\sigma, \mathbf{W}$ and $\mathbf{b}$ are the activation function, weight space and bias vector, respectively, for the encoding transition ϕ, which may or may not be equal to corresponding parameters of reconstruction transition ψ. Through backpropagation of errors, the auto-associative multi-layer perceptron tries to optimally solve the minimization problem formulated as:

$$\phi, \psi = \underset{\phi,\psi}{\arg\min}\, \|\mathbf{z} - (\psi \circ \phi)\,\mathbf{z}\| \quad (7)$$

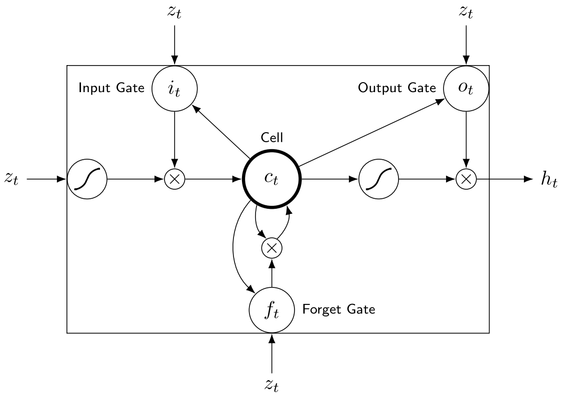


Fig. 6. The LSTM hidden unit with input (i_t), output (o_t) and forget (f_t) gates connected with the cell state c_t; output of the LSTM cell h_t that is passed as an input h_{t+1} to the next hidden unit in a recurrent network.

By reducing a squared Euclidean reconstruction loss, it drives:

$$\mathcal{L}\left(\mathbf{z}, \hat{\mathbf{z}}\right) = \left\|\mathbf{z} - \sigma'\left(\mathbf{W}'\left(\sigma\left(\mathbf{W}\mathbf{z} + \mathbf{b}\right)\right) + \mathbf{b}'\right)\right\|^2 \tag{8}$$

The proposed mapping presented in (6) is stateless in the sense that it would capture sufficiently duration-based characteristics of the normalized likelihood sequence. However, the weight space $\mathbf{W}$ would suffer from the vanishing gradient problem when gradients are back-propagated through time. Since there can be the unpredictable duration of network flow without any transient change, likelihood sequence may exhibit arbitrary chunks of inactivity, thus halting the training of the network well before fully representative learning.

In order to solve this problem, we propose the long short-term memory (LSTM) variant of the autoencoder [44] shown in Figure (6). This can happen by replacing the latent representation $\mathbf{h}$ of (6) with

$$\mathbf{h}_t = \mathbf{o}_t \circ \sigma_h\left(\mathbf{c}_t\right) \tag{9}$$

where $\mathbf{o}_t, \mathbf{c}_t \in \mathbb{R}^h$ are output gate and cell state activation vector of h LSTM units. The symbolic notation $\circ$ denotes relational composition between precedent and antecedent functions. Cell state is governed by a forget gate activation $\mathbf{f}_t$, cell input activation $\tilde{\mathbf{c}}_t$ and update activation $\mathbf{i}_t$ such that

$$\mathbf{c}_t = \left(\mathbf{f}_t \circ \mathbf{c}_{t-1}\right) + \left(\mathbf{i}_t \circ \tilde{\mathbf{c}}_{t-1}\right) \tag{10}$$

so that, the flow of information and retention inside the cell is optimally controlled using (9) and (10), which estimates the LSTM unit using:

$$\mathbf{f}_t = \sigma_g\left(\mathbf{W}_f\mathbf{z}_t + \mathbf{V}_f\mathbf{h}_{t-1} + \mathbf{b}_f\right) \tag{11}$$

$$\mathbf{i}_t = \sigma_g\left(\mathbf{W}_i\mathbf{z}_t + \mathbf{V}_i\mathbf{h}_{t-1} + \mathbf{b}_i\right) \tag{12}$$

$$\mathbf{o}_t = \sigma_g\left(\mathbf{W}_o\mathbf{z}_t + \mathbf{V}_o\mathbf{h}_{t-1} + \mathbf{b}_o\right) \tag{13}$$

$$\mathbf{c}_t = \sigma_h\left(\mathbf{W}_c\mathbf{z}_t + \mathbf{V}_c\mathbf{h}_{t-1} + \mathbf{b}_c\right) \tag{14}$$

where $\mathbf{W} \in \mathbb{R}^{h \times d}, \mathbf{V} \in \mathbb{R}^{h \times h}$ and $\mathbf{b} \in \mathbb{R}^h$ are weights of the input, weights of the recurrent connections and bias parameters required to be learned during the training of the autoencoder with d inputs and h hidden units. The activation functions are given by

$$\sigma_g = S\left(x\right) = \frac{e^x}{e^x + 1} \tag{15}$$

$$\sigma_h = \tanh x = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{16}$$

The network architecture is composed of two types of layers, i.e., LSTM and fully connected layer, as depicted in Figure 4. A number of l LSTM layers comprises h hidden units, where each layer is used to encode the compressed representation of the transformed likelihood stream. The outputs of the hidden units are subsequently connected to a single connected layer with units corresponding to the input packet size. The fully connected layer then produces the reconstructed output $\hat{\mathbf{z}}$. To solve the minimization problem of (7), adaptive learning based stochastic gradient optimization with a mean absolute error loss function is estimated by

$$\epsilon_1 = \frac{\sum_{i=0}^{N-1}\left|z_i - \hat{z}_i\right|}{N} \tag{17}$$

where, the mean squared error is computed by

$$\epsilon_2 = \frac{\sum_{i=0}^{N-1}\left(z_i - \hat{z}_i\right)^2}{N} \tag{18}$$

Algorithm 2 $TrainingAutoencoder\left(\mathbf{z} = \left[z_1, \cdots, z_{N-1}\right]\right)$

```
Initialize inputs
    $L \leftarrow$ length of the sliding window
    $d \leftarrow$ dimension of the data initialized as $d = 1$
    $h \leftarrow$ number of LSTM hidden units $h$
    $b \leftarrow$ batch size $b$
for $l = 0, \cdots, \frac{N}{L} - 1$
    $\mathbf{z}_l \leftarrow$ reshape $\mathbf{z}$ into $\frac{N}{L}$ tensors of shape $(L, d)$
    Pass $z_l$ into a unidirectional RNN layer with $h$ hidden LSTM units
    Compute outputs and states using (9) to (14)
    Solve (7) and (8)
    Compute error using (17)
    $\mathbf{z}_l \leftarrow$ output from a fully connected layer of dimension $d$
    $\mathcal{L}\left(\mathbf{z}, \hat{\mathbf{z}}\right) \leftarrow$ training loss
end for
```

Training of the network is described in Algorithm 2. The initial values are set as $c_0 = h_0 = 0$ before triggering the first training epoch $\hat{z}_{\frac{N}{L}-1}$.

3) Detection Phase: In the detection phase, data features for a short time window $w_j = w_{t_2 - t_1}$ are transformed into the normalized likelihood stream $\mathbf{z}_j$. It then is passed to the recurrent network, where t_1 and t_2 are the initial and final time instants within jth window of interest respectively. A normalized distance metric is calculated by:

$$d_j = \left\|\mathbf{z}_j - \hat{\mathbf{z}}_j\right\|_2 \tag{19}$$

d_j is computed for each window as a decision statistic and compared with a threshold τ to decide between hypothesis $\{\mathcal{H}_0$: normal traffic vector$\}$ and $\{\mathcal{H}_1$: attack traffic vector$\}$ such that:

$$d_j \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\lessgtr}} \tau \tag{20}$$

where τ is the optimal threshold maximizing detection performance metrics such as mean accuracy or minimizing false positives or negatives.

D. Summary of Architectural Workflow

Considering multiple stages and involved complexities of the workflow (Fig. 4), it is necessary to concisely summarise the end-to-end implementation steps. Following steps may serve as a guideline for replicating the proposed architecture.

- Firstly, network flow features are extracted from the raw data. In the case of CAN bus messages, two network features are extracted from Car hacking datase [10]:1) the total number of packets sent by the unique CAN ID in the given time window (3 seconds), and 2) the size of outbound messages from unique ID. Whereas from the second dataset [31] following information is extracted for each channel: total traffic sent, time to live (TTL), bits per second and packet count, The same features are extracted after every three-second time window for both the datasets.
- Iterating over three second time window on the network sequences, the statistical features are extracted here. The features include mean and standard deviation computed from packets or messages data frame size and packets count, as shown in Table I and Table II.
- Compute normalized likelihood transformation following Algorithm 1.
- Training of autoencoder following Algorithm 2.
- Detection phase following (19) and (20).

V. Results and Performance Analysis

A. Experimental Setup

The decision engine is trained in an offline mode using the car hacking and UNSW-NB15 datasets. The experimental setup is explained as follows:

- The datasets were partitioned into 80-20% for training and testing purpose, respectively, for addressing the overfitting problem and ensuring the model will reflect high performance.
- The system was trained on normal data captured from CAN bus network as well as from external networks and was tested for both normal and attack traffic generated from various types of attacks as described in Tables III and IV.
- The proposed deep learning-based IDS can easily run on desktop computers with CPU or GPU support, single-board small computers or any common network gateway or router. Section V-C further explains the implementation choices for training and hyperparameter tuning.

TABLE III

Data Types and Sizes of Car Hacking Dataset

Attack Type	Total Messages	Normal Messages	Injected Messages
Normal	988,987	988,872	
DoS attack	3,665,771	3,078,250	587,521
Fuzzy attack	3,838,860	3,347,013	491,847
Gear Spoofing	4,443,142	3,845,890	597,252
RPM guage spoofing	4,621,702	3,966,805	654,897

TABLE IV

Data Types and Sizes of UNSW-NB15 Dataset

Attack Type	Number of Packets
Normal	677785
Exploits attack	5408
Generic attack	7522
DoS attack	1167
Fuzzer attack	5,051
Recon attack	1759

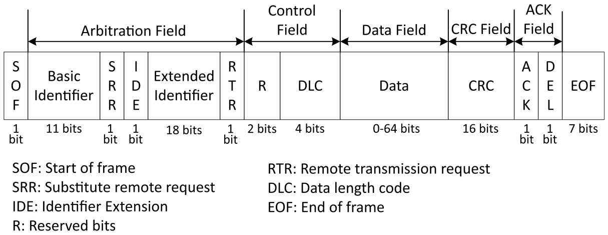


Fig. 7. Structure of CAN 2.0B message frame.

B. Datasets

The two datasets used for evaluation of the proposed IDS are: 1) car hacking dataset [10] for evaluating detection of anomalies in in-vehicles communication and 2) UNSW-NB15 dataset [31] for anomaly detection generated from external networks. The description of the two datasets is as follows:

1) The Car Hacking Dataset: The structure of CAN frame is shown in Figure 7. The dataset is available as a comma-separated value file from [10]. Table III shows the details about the car hacking dataset. This data was constructed by capturing logs of CAN traffic through OBD-II port of a real vehicle [10]. The logs were taken while the attacks were in progress.

The attack types of the dataset include fuzzy attack, DoS attack, spoofing the RPM gauge and spoofing the drive gear. Each dataset contains 30-40 minutes of CAN traffic. There were 300 instructions of message injection and each attack conducted for 3-5 seconds. The important features are extracted for use in the proposed IDS: CAN ID, COUNT, DATA[0], DATA[1], DATA[2], t DATA[3],DATA[4], DATA[5], DATA[6], DATA[7].

2) The UNSW-NB15 Dataset: The dataset is available at [31] as a comma-separated value file. This dataset was generated for evaluation of NIDS by creating a synthetic laboratory environment. The IXIA PerfectStorm tool was used to generate

TABLE V
HYPERPARAMETERS OF LSTM MODEL FOR CAN AND UNSW DATASET

	L	h	b	$\mathcal{L}(\mathbf{z}, \hat{\mathbf{z}})$	*epochs*
CAN	10	50	100	ϵ_2	500
UNSW	10	100	50	ϵ_2	2000

a hybrid of normal and attack traffic. The dataset includes network flow data of nine major families of attacks. The Bro-IDS and Argus tools were used for extracting desired features from the pcap files of the network data. Table IV shows the details about the dataset used to evaluate the proposed IDS. 11 important features are selected from a total of 49 network flow attributes which contribute in the effective construction of normal and attack behaviors of the traffic, which include *source IP, destination IP, source port and destination port, duration, source bytes, destination bytes, source TTL, destination TTL, source load, destination load, source packets, and destination packets.*

C. Detection Performance

The proposed model was implemented using python 3.8.2 and tensor-flow application programming interface. The experiments were performed on a computer system with graphical processing unit support. The details of the training process, hyperparameter tuning, threshold optimization and detection results are described below.

1) Training and Hyperparameter Tuning: The final set of hyperparameters and trajectory of training loss for both datasets is depicted in Table V and Figure 8, respectively. For both datasets, the sliding window size of length $L = 10$ was found suitable to facilitate latent encoding and subsequent reconstruction. The UNSW dataset took a lot more epochs than CAN to reduce average training and validation losses to acceptable limits. The quick grid search showed that the batch size b between 10 to 100 was suitable for training, however, $b = 100$ and $b = 50$ for the CAN and UNSW datasets, respectively, were chosen considering the optimal time taken for training. Being spatio-temporally more complex, the normalized likelihood transformation of the UNSW dataset required considerably more hidden LSTM units ($h = 100$) as well as four times more epochs for a good reconstruction of input batches while ensuring that model is not over-fitting, thereby degrading accuracy. The trajectory of training loss shows that even after the considerable reduction in loss, mean squared error ϵ_2 for both datasets remained in orders of magnitude less than mean absolute error ϵ_1.

2) Selection of Optimal Threshold: The autoencoder successfully reconstructs the normalized likelihood sequence of normal, i.e, legitimate network traffic, therefore reconstruction error in case of all the unseen attack vectors is distinguishable from the normal flow. The error surfaces in the case of both the datasets are linearly separable with varying degrees of error, the accuracy of which is dependent upon the choice of threshold τ. To implement the decision statistic of (19) and (20), the optimal threshold τ_{opt} is found employing grid search through receiver operating characteristics (ROC) as shown in Figure 9. The detection probabilities were calculated for the entire range of $T = \tau \in \lceil 0.05, 0.95 \rceil$ and ROC curves were obtained by estimating the true positive rate (TPR) and false positive rate (FPR), respectively, and given by:

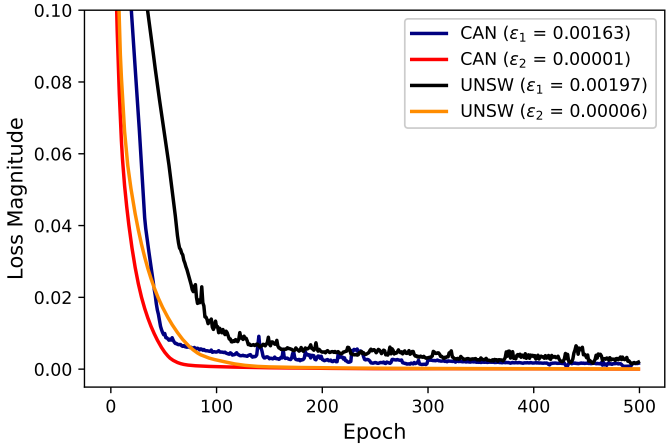


Fig. 8. Trajectory of the loss during epochs of training the autoencoder for CAN and UNSW datasets.

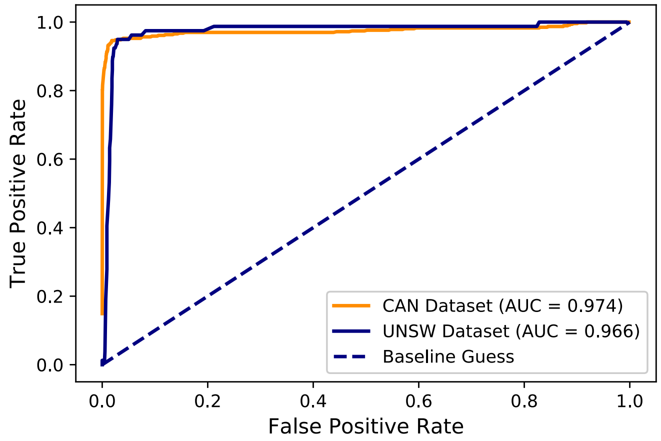


Fig. 9. Grid search for finding optimal threshold τ.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{22}$$

where true positive (TP) and true negative (TN) are the normal and attack vectors classified correctly, respectively. On the other hand, being a binary classification problem with all attack vectors lumped as illegitimate, false negative (FN) means those among them which are erroneously miss-classified by the classifier as legitimate normal traffic. Lastly, the false positive (FP) are those normal sequences that are wrongly classified as anomalous.

Before choosing τ_{opt}, the area under the ROC curve (AUC) was chosen as a sufficient confidence metric to gauge the probability in which the proposed model ranks a random normal vector higher than a random attack vector. This general confidence metric is important to provide necessary heuristics since it measures the quality of the models' predictions irrespective of the chosen value of τ. Both models perform generally similar to the one trained and validated on the normal vectors of the CAN dataset having a little edge as compared to the model trained on the normal vectors of the UNSW dataset.

TABLE VI

DETECTION RESULTS ON CAN AND UNSW DATASETS WITH OPTIMAL THRESHOLDS MAXIMIZING VARIOUS PERFORMANCE METRICS

	$\arg\max_{\phi}$	$\phi_{\text{precision}}$	ϕ_{recall}	ϕ_{F1}	ϕ_{accuracy}
CAN	ϕ_{accuracy}	0.99	1.00	0.99	0.99
	ϕ_{F1}	0.99	1.00	0.99	0.99
	ϕ_{gmean}	1.00	0.98	0.99	0.98
UNSW	ϕ_{accuracy}	0.99	0.98	0.99	0.975
	ϕ_{F1}	0.94	1.00	0.97	0.944
	ϕ_{gmean}	1.00	0.97	0.98	0.969

The value of τ_{opt} can be subsequently chosen with the consideration of maximizing three different performance metrics ϕ, i.e.,

$$\phi_{\text{accuracy}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{23}$$

$$\phi_{\text{F1}} = 2 \times \frac{\phi_{\text{precision}} \times \phi_{\text{recall}}}{\phi_{\text{precision}} + \phi_{\text{recall}}} \tag{24}$$

$$\phi_{\text{gmean}} = \sqrt{\text{TPR} \times (1 - \text{FPR})} \tag{25}$$

where $\phi_{\text{precision}}$ is the positive predictive value of the classifier and ϕ_{recall} is the classifier sensitivity or TPR as given in (21). Consequently, optimal threshold τ_{opt} is given by the relationship

$$\tau_{\text{opt}} = \arg\max_{\phi} T\,(\phi) \tag{26}$$

where $\phi \in \left(\phi_{\text{accuracy}}, \phi_{\text{F1}}, \phi_{\text{gmean}}\right)$ computed using (23), (24) and (25), respectively.

3) Detection Results and Explanations: Table VI lists the classification results for the attack class with choice of τ_{opt} in all three contexts of maximizing each performance metric, i.e., ϕ_{accuracy}, ϕ_{F1} and ϕ_{gmean}.

Both models perform best in terms of accuracy of correctly classifying $\mathcal{H}_1$ categorizing the attack class with $\phi_{\text{accuracy}} = 0.99$ and $\phi_{\text{accuracy}} = 0.975$ for the CAN and UNSW datasets, respectively. The precision of 100% is observed in both cases where τ_{opt} is chosen to maximize the geometric mean metric ϕ_{gmean}, which effectively means the difference between true positive and false positive rates. Maximizing ϕ_{gmean} implies that the choice of τ_{opt} which seeks a balance between sensitivity and specificity of the classifier since ϕ_{gmean} is, in fact, the product of both parameters. Since there is an inherent imbalance in input classes and absolutely no analytical justification of training the autoencoders on attack vectors, τ_{opt} with maximum ϕ_{gmean} implies confidence in the utility of the classifier in heterogeneous environments, with multifarious attack vectors of network flows.

When τ_{opt} is chosen so as to maximize ϕ_{F1}, an accuracy of $\phi_{\text{accuracy}} = 0.99$ and $\phi_{\text{accuracy}} = 0.944$ is observed for the CAN and UNSW dataset, respectively. Since ϕ_{F1} is a metric balancing $\phi_{\text{precision}}$ and ϕ_{recall}, it provides a rational measure putting classifier's exactness and completeness in equilibrium. It is a slightly low average accuracy in the case of the UNSW dataset, which means that latent representation of flow dynamics is hard to learn by the designed autoencoder model. However, $\phi_{\text{precision}} = 0.94$ and $\phi_{\text{recall}} = 1.00$ show that though there are false positives, no attack vectors were miss-classified as normal observations.

D. Comparison and Discussion of the Proposed IDS

With the emergence of smart cities and the development of new telecommunication technologies for AVs, there is great attention to protect them against cyber-attacks and in order to avoid physical damage and save human lives. AVs are complex systems that include various hardware and software elements that generate heterogeneous data sources from physical devices and network connections. The proposed IDS is compared with the other four techniques, in the evaluation terms of precision, recall, f1 and accuracy, on each of the Car Bus (i.e., CAN) and UNSW-NB15 datasets, as listed in Table VII. on the CAN dataset, the proposed deep learning-based IDS achieved higher detection accuracy than the techniques of Naive Bayes (NB), Artificial Neural Network (ANN), Support Vector Machine (SVM) and Decision Tree (DT) [35]. Similarly, on the UNSW-NB15 dataset, the proposed IDS accomplished higher performance than the techniques of Artificial Immune System (AIS), Euclidean Distance Map (EDM), Filtered-based SVM (FSVM) and Geometric Area Analysis (GAA) [22].

To give a clear discussion about the better performance of the proposed IDS compared with the other techniques, we explain that the proposed system can effectively discover cyber-attacks from AVs and their network traffic. The intrusion detection models listed in Table VII suffer from handling large-scale data collected from the physical and network element for achieving high detection accuracy and low false alarm rates. The proposed IDS has several advantages that address the limitations of existing IDSs in AVs. It includes a feature extraction module that filters and processes high dimensionality of data and extracts more representative features that contain patterns of normal behaviour. This module has a great impact to remove any data redundancy and obtain the most significant features, handling the challenge of data heterogeneity, and assisting in improving the performance of the deep learning-enabled detection method.

The proposed detection method using the architecture of LSTM Autoencoder is also well-designed to handle time-series data (e.g., the Car Bus and UNSW-NB15 dataset) and encode them to adopt the most important features. The LSTM technique then employs the most import features as inputs to learn hidden patterns of legitimate and suspicious events. Then, the decision engine can precisely classify attack and normal behaviors, accomplishing high detection accuracy, low false alarm rates in real-time evaluations. However, the proposed system still has a limitation of accurately identifying attack categories of AVs and their networks, as the kernel function and optimizer of the proposed detection method can considerably classify binary classes (i.e., normal or attack events). This limitation will require new adapted functions that can handle the multiclass issue, including normal and attack types such as DoS and DDoS.

VI. CONCLUSION AND FUTURE WORK

The IoVs are exposed to various types of cyber-attacks because of their ubiquitous nature and poor security implementation in the technologies used. IDSs offer effective solutions to protect IoVs from multiple types of attacks. This work

TABLE VII

COMPARSIONS OF PROPOSED IDS WITH OTHER TECHNIUQES ON BOTH DATASETS

CAN Dataset					**UNSW-NB15 dataset**				
Techniques	**Precision**	**Recall**	**F1**	**Accuracy**	**Techniques**	**Precision**	**Recall**	**F1**	**Accuracy**
NB	0.76	0.74	0.76	0.73	AIS	0.89	0.86	0.87	0.85
ANN	0.83	0.81	0.82	0.82	EDM	0.92	0.91	0.91	0.90
SVM	0.92	0.89	0.90	0.91	FSVM	0.94	0.92	0.93	0.92
DT	0.94	0.93	0.93	0.92	GAA	0.96	0.94	0.95	0.93
Our IDS	0.99	1.00	0.99	0.99	Our IDS	1.00	0.97	0.98	0.96

has focused on developing an IDS model that learns the behavior of normal network traffic in IVNs and detects attacks through deviations from the learned latent representation of the messages and network flows. We proposed LSTM autoencoder based IDS for detecting abnormal events from IoVs. The proposed scheme integrates and combines the strength of statistical features of the network behavior as well as the robust learning mechanism of LSTM autoencoder to develop an effective learning model of normal traffic flow in IoVs.

To evaluate our proposed IDS we tested our IDS on two benchmark datasets, i.e car hacking dataset for IVNs and UNSW-NB 15 dataset for external network communications. The evaluation results demonstrate that the proposed model produced excellent results in terms of accuracy, F-1 score and detection rate. The overall average accuracy of the proposed IDS is 99% and 98% for car hacking and UNSW-NB-15 datasets, respectively.The major advantages of our proposed scheme in comparison to other related works is that, unlike other studies related to IDS for IVNs, which generally implement IDS for either CAN bus or for external communication networks, our proposed models for IVNs and external communications are designed on an integrated architectural backbone which works for both type of cyber-attacks on IoVs with little variation. Another significance of the proposed architecture is its detection of multiple type of attack vectors instead of one or few types, as handled by the IDSs in other similar work.

We briefly discussed various options for placement of an IDS in IVNs, however there is no published work on evaluation of IDS placement options with advantages and disadvantages of each, and thus it still is a challenge and an open area of research. Another challenge and potential area of research is proposing an IDS, which can work in large-scale IoVs environment, as well as, is robust and credible for detecting complex attack scenarios including zero-day attacks.

REFERENCES

[1] C. Lai, R. Lu, D. Zheng, and X. Shen, "Security and privacy challenges in 5G-enabled vehicular networks," *IEEE Netw.*, vol. 34, no. 2, pp. 37–45, Mar. 2020.

[2] M. Obaidat, M. Khodjaeva, J. Holst, and M. B. Zid, "Security and privacy challenges in vehicular ad hoc networks," in *Connected Vehicles in the Internet of Things*. Cham, Switzerland: Springer, 2020, pp. 223–251.

[3] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Black Hat USA, 2015, p. 91.

[4] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," Briefing, Black Hat USA, 2017, pp. 1–16.

[5] C. Bernardini, M. R. Asghar, and B. Crispo, "Security and privacy in vehicular communications: Challenges and opportunities," *Veh. Commun.*, vol. 10, pp. 13–28, Oct. 2017.

[6] W. Wu *et al.*, "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 919–933, Mar. 2020.

[7] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Netw.*, vol. 31, no. 5, pp. 50–58, Sep. 2017.

[8] D. Kosmanos *et al.*, "A novel intrusion detection system against spoofing attacks in connected electric vehicles," *Array*, vol. 5, Mar. 2020, Art. no. 100013.

[9] Y. Sun *et al.*, "Attacks and countermeasures in the Internet of vehicles," *Ann. Telecommun.*, vol. 72, nos. 5–6, pp. 283–295, 2017.

[10] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in *Proc. 16th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2018, pp. 1–6.

[11] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLoS ONE*, vol. 11, no. 6, Jun. 2016, Art. no. e0155781.

[12] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," in *Proc. 15th Annu. Conf. Privacy, Secur. Trust (PST)*, Aug. 2017, pp. 57–5709.

[13] T. Marsden, N. Moustafa, E. Sitnikova, and G. Creech, "Probability risk identification based intrusion detection system for SCADA systems," in *Proc. Int. Conf. Mobile Netw. Manage.* Cham, Switzerland: Springer, 2017, pp. 353–363.

[14] N. Moustaf and J. Slay, "Creating novel features to anomaly network detection using DARPA-2009 data set," in *Proc. 14th Eur. Conf. Cyber Warfare Secur.* New York, NY, USA: Academic, 2015, pp. 204–212.

[15] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and K.-K. R. Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5110–5118, Aug. 2020.

[16] J. P. Liu, O. F. Beyca, P. K. Rao, Z. J. Kong, and S. T. S. Bukkapatnam, "Dirichlet process Gaussian mixture models for real-time monitoring and their application to chemical mechanical planarization," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 208–221, Jan. 2017.

[17] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," *Inf. Secur. J., A Global Perspective*, vol. 25, nos. 1–3, pp. 18–31, Apr. 2016.

[18] H. Sedjelmaci, S. M. Senouci, and M. Al-Bahri, "A lightweight anomaly detection technique for low-resource IoT devices: A game-theoretic methodology," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[19] S. Rathore and J. H. Park, "Semi-supervised learning based distributed attack detection framework for IoT," *Appl. Soft Comput.*, vol. 72, pp. 79–89, Nov. 2018.

[20] A. K. Gupta and S. Nadarajah, *Handbook of Beta Distribution and Its Applications*. Boca Raton, FL, USA: CRC, 2004.

[21] N. Moustafa, J. Slay, and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Trans. Big Data*, vol. 5, no. 4, pp. 481–494, Dec. 2019.

[22] N. Moustafa, K.-K.-R. Choo, I. Radwan, and S. Camtepe, "Outlier Dirichlet mixture mechanism: Adversarial statistical learning for anomaly detection in the fog," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 8, pp. 1975–1987, Aug. 2019.

[23] N. Moustafa, G. Creech, E. Sitnikova, and M. Keshk, "Collaborative anomaly detection framework for handling big data of cloud computing," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6.

[24] T. Chandak, S. Shukla, and R. Wadhvani, "An analysis of 'A feature reduced intrusion detection system using ANN classifier' by Akashdeep et al. expert systems with applications (2017)," *Expert Syst. Appl.*, vol. 130, pp. 79–83, Sep. 2019.

[25] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech, "Privacy preservation intrusion detection technique for SCADA systems," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6.

[26] Q. Li, F. Wang, J. Wang, and W. Li, "LSTM-based SQL injection detection method for intelligent transportation system," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4182–4191, May 2019.

[27] K. Zhu, Z. Chen, Y. Peng, and L. Zhang, "Mobile edge assisted literal multi-dimensional anomaly detection of in-vehicle network using LSTM," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4275–4284, May 2019.

[28] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in Internet of vehicles," 2019, *arXiv:1910.08635*. [Online]. Available: http://arxiv.org/abs/1910.08635

[29] A. Diro and N. Chilamkurti, "Leveraging LSTM networks for attack detection in Fog-to-Things communications," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 124–130, Sep. 2018.

[30] W. Zhang *et al.*, "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, 2018.

[31] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[32] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 130–139.

[33] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected car cybersecurity," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–7.

[34] M. Aloqaily, S. Otoum, I. A. Ridhawi, and Y. Jararweh, "An intrusion detection system for connected vehicles in smart cities," *Ad Hoc Netw.*, vol. 90, Jul. 2019, Art. no. 101842. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1570870519301131

[35] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Veh. Commun.*, vol. 21, Jan. 2020, Art. no. 100198.

[36] L. Nie, Z. Ning, X. Wang, X. Hu, Y. Li, and J. Cheng, "Data-driven intrusion detection for intelligent Internet of vehicles: A deep convolutional neural network-based method," *IEEE Trans. Netw. Sci. Eng.*, early access, Apr. 27, 2020, doi: 10.1109/TNSE.2020.2990984.

[37] H. H. R. Sherazi, R. Iqbal, F. Ahmad, Z. A. Khan, and M. H. Chaudary, "DDoS attack detection: A key enabler for sustainable communication in Internet of vehicles," *Sustain. Comput., Informat. Syst.*, vol. 23, pp. 13–20, Sep. 2019.

[38] L. Song, G. Sun, H. Yu, X. Du, and M. Guizani, "FBIA: A fog-based identity authentication scheme for privacy preservation in Internet of vehicles," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5403–5415, May 2020.

[39] A. H. Sodhro, G. H. Sodhro, M. Guizani, S. Pirbhulal, and A. Boukerche, "AI-enabled reliable channel modeling architecture for fog computing vehicular networks," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 14–21, Apr. 2020.

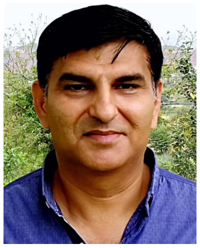
[40] P. Shrivastava, M. S. Jamal, and K. Kataoka, "EvilScout: Detection and mitigation of evil twin attack in SDN enabled WiFi," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 1, pp. 89–102, Mar. 2020.

[41] G. Kornaros *et al.*, "Towards holistic secure networking in connected vehicles through securing CAN-bus communication and firmware-over-the-air updating," *J. Syst. Archit.*, vol. 109, Oct. 2020, Art. no. 101761.

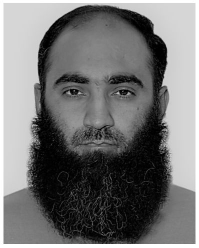
[42] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[43] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biol. Cybern.*, vol. 59, nos. 4–5, pp. 291–294, Sep. 1988.

[44] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.



Javed Ashraf received the B.E. and M.S. degrees in computer software engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2002 and 2010, respectively, where he is currently pursuing the Ph.D. degree. He has a vast experience of 17 years in IT industry and academia. His main research interests include natural language processing, data science, software-defined networks security, IoT security, and IoVs security using machine learning and deep learning techniques.

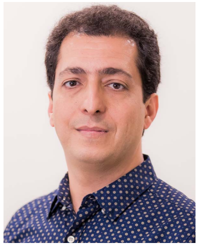


Asim D. Bakhshi received the B.E. degree in electrical and computer engineering from the National University of Sciences and Technology (NUST), Islamabad, in 1996, and the M.S. and Ph.D. degrees in computer engineering from the University of Engineering and Technology (UET), Lahore, in 2002 and 2012, respectively. He is currently an Assistant Professor at NUST. His research interests include biomedical signal processing and image processing. He is currently focusing on the applications of machine learning architectures in the areas of anomaly detection in enterprise networks and cyber-physical systems.

Nour Moustafa (Senior Member, IEEE) received the B.S. and M.S. degrees in computer science from the Faculty of Computer and Information, Helwan University, Egypt, in 2009 and 2014, respectively, and the Ph.D. degree in cyber security from the University of New South Wales (UNSW), Canberra in 2017. His areas of interests include cyber security, in particular, network security, host- and network-intrusion detection systems, statistics, deep learning, and machine learning techniques. He is currently a Lecturer in cyber security and the Coordinator of the Postgraduate Cyber Program at the School of Engineering and Information Technology (SEIT), UNSW Canberra.

Hasnat Khurshid received the Ph.D. degree in electrical engineering from the National University of Sciences and Technology in 2015. He has a versatile experience of 15 years in industry and academia. His prime area of expertise is intelligence extraction from multidimensional data using feature extraction and machine learning techniques. His skill-set includes a vast variety of mathematical, algorithmic, and software tools for research, development, and numerous industrial applications.

Abdullah Javed received the B.S. degree in computer science from the Sir Syed Center for Advanced Studies in Engineering (CASE), Institute of Technology, Islamabad, Pakistan, in 2020. He is currently working as a Research Associate with the National University of Sciences and Technology (NUST), Islamabad. His main research interests include data science, machine learning, and IoT security.



Amin Beheshti received the B.S. and M.S. degrees (Hons.) in computer science and the Ph.D. degree in computer science and engineering from UNSW Sydney. He is currently the Director of the AI-Enabled Processes (AIP) Research Centre and the Head of the Data Analytics Research Laboratory, Department of Computing, Macquarie University. He is also a Senior Lecturer in data science with Macquarie University and an Adjunct Academic in computer science with UNSW Sydney. He is the leading author of the book *Process Analytics*, coauthored with other high-profile researchers in UNSW and IBM research, recently published by Springer.