

XAIEnsembleTL-IoV: A new eXplainable Artificial Intelligence ensemble transfer learning for zero-day botnet attack detection in the Internet of Vehicles

Yakub Kayode Saheed^{*}, Joshua Ebere Chukwuere

Department of Information Systems, North-West University, Mahikeng, South Africa

ARTICLE INFO

Keywords:

Internet of vehicle
eXplainable artificial intelligence (XAI)
CAN-Hacking
Zero-day
Botnet
Barnacle mating optimizer
ResNet
Inception
Inception ResNet
MobileNet
N-BaIoT

ABSTRACT

The Internet of Vehicles (IoV) is a network of interconnected vehicles that use modern communication technologies to communicate with each other and the surrounding infrastructure. The IoV is a novel network that experiences a continuous emergence and evolution of various forms of attacks. This research addresses the growing challenge of detecting zero-day botnet attacks within the IoV, a complex network that enhances real-time communication among vehicles and surrounding infrastructure to improve traffic management, safety, and driving experiences. Traditional machine learning-based intrusion detection systems (IDS) for IoV face two key limitations: the requirement for large labeled datasets and the “black box” issue, where the reasoning behind model decisions is not transparent, reducing user and stakeholder confidence. To solve these problems, this research proposes an eXplainable Artificial Intelligence (XAI) Ensemble Transfer Learning (TL) model specifically for detecting zero-day attacks in IoV. The proposed model integrates deep Shapley Additive Explanations (SHAP), providing transparency and making decisions understandable to cybersecurity professionals. Additionally, the model employs hybrid bidirectional long-short-term memory with autoencoders (BiLAE) to reduce the dimensionality of IoV network traffic, improving computational efficiency. It also uses Barnacle Mating Optimizer (BMO) to optimize the hyper-parameters of deep learning models such as ResNet, Inception, Inception ResNet, and MobileNet Convolution neural network-transfer learning architecture (CNN-TL), enhancing detection capabilities without needing vast amounts of labeled data. Experimental results showed that the model performed with an accuracy of 100 %, precision of 100 %, recall of 100 %, F1-score of 100 %, and Matthew Correlation Coefficient of 100 % in binary-class situations for internal vehicular (CAN) networks and achieved 99.88 % accuracy and similarly high metrics in multi-class scenarios for external vehicular networks (N-BaIoT). Compared to state-of-the-art techniques, the model proved to be more effective in detecting zero-day botnet attacks, reducing reliance on large datasets. Unlike traditional black-box models, the XAI component of the ensemble model offers insight into the decision-making process. It allows network administrators and security experts to understand how specific patterns in the data contribute to detection, making the system more transparent. The solution is highly adaptable and scalable for real-time application, designed to operate efficiently even on IoV gateway electronic control units with limited computational power.

1. Introduction

As network connection technology advances, particularly with the 5G technology, the level of vehicle networking is increasing [1]. Tesla, the renowned leader in the intelligent automobile sector, achieved sales of 484,500 electric vehicles during the last quarter of 2023, amounting to a total of 1.8 million units delivered throughout the entire year. The year-end performance exceeded experts' expectations, with a 20 %

increase in sales compared to the same period in 2022, and a faster growth rate compared to early in 2023. The Chinese company BYD achieved a remarkable sales milestone in the last quarter of 2023 by selling a total of 526,000 vehicles powered solely by batteries [2]. Alibaba has developed automotive networking solutions based on AliOS, enabling intelligent assistants, on-board navigation, and other uses. All of them offer vehicle status updates, remote diagnosis, preventive maintenance, and remote access using mobile networks. The IoV

^{*} Corresponding author.

E-mail addresses: yakubu.saheed@aun.edu.ng (Y.K. Saheed), josh.chukwuere@nwu.ac.za (J.E. Chukwuere).

<https://doi.org/10.1016/j.rineng.2024.103171>

Received 30 July 2024; Received in revised form 6 October 2024; Accepted 14 October 2024

Available online 18 October 2024

2590-1230/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

primarily comprises of external vehicular networks and intra-vehicle networks [3]. IVNs incorporate a growing quantity of electronic control units (ECUs) to implement diverse capabilities [4]. A controller area network (CAN) bus connects the various ECUs within a vehicle, facilitating message transmission and task execution [5]. However, external networks establish connections between contemporary automobiles and the surrounding environment through V2X technology [6]. V2X technology enables contemporary automobiles to establish communication with other vehicles, road users, and roadside infrastructures [7]. Nevertheless, as the IoV continues to advance, it also brings to light security concerns like susceptibility to zero-day botnet attacks, remote intrusions, and malicious manipulation [8].

In 2024, several teams of highly skilled hackers participated in the inaugural Pwn2Own hacking tournament, which only targeted the automotive industry. At the Automotive World conference in Tokyo, a group of hackers convened, with a total of over \$1 million in bounty rewards available. Tesla, widely recognized as the leading brand in the electric vehicle industry, saw two successful hacking incidents perpetrated by the same group of hackers known as Team Synacktiv. Notably, Team Synacktiv emerged as the overall competition winners, receiving a substantial sum of \$450,000. Security researchers discovered a vulnerability in a server of Calamp, a telematics firm that provides vehicle security and tracking services, in 2018. This vulnerability enabled hackers to gain unauthorized access to account data and even exert direct control over the targeted vehicle [1].

While some common security measures, like cryptography and authentication, work well for other networks, they might not work well with CANs or could mess up the timing needed for CAN communications in intervehicle networks [9]. Therefore, intrusion detection systems (IDSs) have become a crucial element in contemporary IoV to detect and pinpoint malevolent attacks on vehicular networks [10]. Incorporating IDSs into external networks is crucial for identifying and detecting malicious attacks that have the potential to bypass firewalls and authentication systems [11]. Despite earlier successful attempts at designing IDSs, intrusion detection remains a tough challenge due to the large amount of network traffic data, multiple network properties, and diverse cyber-attack methods [12,13]. However, the enhanced connection and ease of access of vehicular networks have expanded the potential targets for cyber-attacks on contemporary vehicles [13]. In addition, the processing of CAN packets does not include any authentication or encryption schemes due to their low length [14]. Cyber-attackers are able to inject malicious messages into IoV networks due to the absence of essential security mechanisms. Hence, it is imperative to create IDSs in order to safeguard IoV networks and intelligent vehicles by detecting cyber-attacks [15]. The IDS in the IoV is depicted in Fig. 1. Specifically, the architecture of the machine learning-based services in the IoV can be implemented by vehicle manufacturers or security providers who are involved in the IDS. Once a vehicle is manufactured and exits the factory, the initial ML model is implemented.

However, the original model's performance will significantly decrease due to the advent of new zero-day attacks that have a different feature distribution. To effectively handle these zero-day attacks, it is necessary to update the intrusion detection model that relies on ML. In

addition, it is imperative to transfer network traffic data from the actual environment to the IoV cloud. Subsequently, a new detection model must be trained using this new data. The updated model will then be implemented on vehicles to identify and counteract new threats. Nevertheless, the process of labelling data for servers demands substantial effort and time, making it unfeasible to acquire a sufficient amount of labeled data within a limited timeframe to train a new detection model for IoV. Transfer learning [1,16], would effectively addresses the aforementioned issues.

The use of an IDS is essential for efficient detection of potentially malicious network connections [17]. Contrary to conventional IDS systems used for the Internet, vehicle networks have a limited set of recognized attack signatures. In addition, IDS for vehicles necessitate a high level of accuracy, as any erroneous detection of false positives might have a significant impact on the driver's safety [18]. In order to address this issue, we propose the implementation of an Explainable (XAI) ensemble transfer learning IDS model for both in-vehicle and external vehicle networks. This system utilizes Bidirectional Long Short-Term Memory autoencoder (BiLAE) to decrease the dimensionality of extensive IoV network traffic data through the encoding phase of BiLAE. Additionally, it incorporates convolution neural network-transfer learning (CNN-TL) architectures by combining ResNet, InceptionResNet, Inception and MobileNet. Soft ensemble learning employed the average of probability, while the Barnacle mating optimizer was utilized to optimize the hyperparameters of each transfer learning technique, including ResNet, InceptionResNet, Inception and MobileNet. As far as we know, this proposed DL has not been used before to decrease the number of features in the CAN intra-vehicle network and N-BaIoT external vehicular dataset. Nevertheless, the TL approach has not been previously employed to categorize the latent space representation of IoV network traffic aspects. TL has the ability to identify and classify unfamiliar threats by solely analyzing normal data. This research primarily presents the following contributions:

- We proposed a new framework for detecting cyber-attacks in both the internal vehicle network and external networks. To enhance its effectiveness, this framework combines BiLAE, CNN-TL architectures, average of probability ensemble learning, and Barnacle mating optimization (BMO).
- We propose a BiLAE with a single hidden layer to reduce the dimensionality of features. This method cuts down on the number of dimensions in large amounts of external vehicular network and intra-vehicle network traffic data. It also creates a low-dimensional latent-space feature representation at the hidden layer without losing any useful network information.
- We design the Deep SHAPley additive eXplanation to elucidate the results of IDS in IoV networks. Previous SHAP methods have provided explanations for AI models in various natural language processing and computer vision scenarios, but our research introduces the first application of Deep SHAP for botnet identification in the IoV.
- The efficiency of BiLAE-ensemble CNN-TL (ResNet, InceptionResNet, Inception and MobileNet) in binary classification and multi-class classification scenarios is validated through extensive experiments

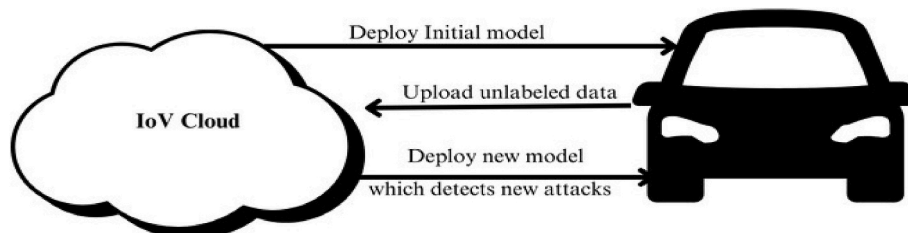


Fig. 1. An architecture for upgrading the intrusion detection model in the IoV based on ML technique.

conducted on the CAN intra-vehicular network data and N-BaIoT external vehicular network data. We compare its performance to various ML and contemporary DL frameworks in the literature. The comparison of performance shows that the proposed XAI Ensemble transfer learning model is better than other frameworks, proving that it can find zero-day botnets in the IoV.

- As far as we know, our research is the first to attempt to design the BMO for enhancing the hyperparameters of ResNet, Inception ResNet, Inception, and MobileNet CNN-TL architectures. The proposed research aims to enhance the development of optimized, efficient, and accurate zero-day botnet intrusion detection for the IoV.
- We developed the proposed XAI ensemble model attack detection framework which consists of XAI module, soft voting ensemble module, to assist cyber security professionals by easing the burden of extensive investigations and ensuring the security of IoV and the safety of drivers against cyber-attacks.

The subsequent sections of the paper are organized in the following manner. Section 2 contained a discussion of the related research. Section 3 outlines the proposed XAI ensemble transfer learning, while Section 4 presents the Bidirectional Long-Short term memory autoencoder, transfer learning and XAI. Section 5 presents the discussion and experimental setup. Section 6 encompasses the concluding remarks and prospects for future research.

2. Related work

The authors [19] introduced a DNN based IDS designed specifically for securing In-Vehicle networks. The experiment's findings demonstrated that the model achieved an average detection ratio of over 98 % on the CAN bus. The researchers [20] introduced a model for classifying intrusions on vehicle networks using KNN and SVM algorithms to detect CAN intrusions. The experimental results showed that KNN achieved an accuracy of 98.3 %, recall of 96.9 %, precision of 97.4 %, and F1 of 98.2 % while SVM gave an accuracy of 97.3 %, recall of 96.4 %, precision of 96.5 % and F1 score of 98.3 %. However, the model employed two data sets, one for fuzzy attack and other one for DoS attack without factoring zero-day botnet attack. The authors [21] presented a new approach for gathering the real-time model of the CAN bus. The authors obtained an accuracy of 87.21 %, a recall of 86.6 %, and a f1-score of 92.1 % without taking into account the MCC and precision. Furthermore, the study did not take into account the explain ability of the model, duration of training and testing. The authors [22] presented a DCNN to safeguard the CAN bus of the vehicle. The proposed model achieved an accuracy rate of 99.93 %, a recall rate of 99.84 %, and an F1-score of 99.91 %. However, the work did not take into account the MCC and precision. Furthermore, the proposed model is a complex and opaque model, making it challenging to comprehend its interpretability. The authors [10] established an IDS that utilizes a generative adversarial network to detect Fuzzy, DoS, Gear, and RPM assaults in the traffic of a CAN bus network. The outcomes of several experiments indicate that all four attacks were identified with a commendable accuracy with DoS attack achieved an accuracy of 97.9 %, precision of 96.8 %, AUC of 99 %, and DR of 99.6 % hence showcasing the resilience of the model. The authors [23] introduced an IDS that utilizes a RNN with LSTM. The chosen model exhibited a superior validation accuracy score of 98.76 %, particularly in identifying spoofing attacks inside the CAN network traffic. This finding serves as a stimulus for further investigation in order to spot various forms of cyber-attacks. The authors [24] suggested an IDS based on a Long Short-Term Memory neural network. The proposed IDS has the ability to identify many types of assaults on the CAN bus network, including fuzzing, spoofing and DoS attacks. The experimental analysis revealed an accuracy of 99.99 %, recall of 99.94 %, F1 of 99.97 %, and AUC of 100 %. The authors [25] propose a DL-based IDS for ITS, specifically designed to detect suspicious network activity in In-vehicle

networks. An autoencoder system based on a DL architecture, namely LSTM, is developed to identify intrusive events occurring at the central network gateways of autonomous vehicles. The experimental analysis demonstrated a precision rate of 99.0 %, a recall rate of 100 %, an F1-score of 100 %, and an accuracy rate of 99.0 % when applied to CAN bus data. When connected to an external vehicular network, the precision was 100 %, the recall was 97.0 %, the F1-score was 98.0 %, and the accuracy was 96.0 % on the UNSW-NB15 dataset. The authors [26] introduced an advanced safeguarding method for self-driving vehicles, specifically designed to ensure the security of external communications in both semi-autonomous and self-driving vehicles. The proposed hybrid detection method employs BPNNs to identify DoS, a prevalent form of attack in vehicular adhoc networks. The experiment's results demonstrated a recall rate of 99.06 % and an accuracy rate of 99.05 %. The authors [27] proposed a mechanism utilizing a FFNN for the purpose of intrusion detection. The performance of the proposed model is assessed on the CICIDS2017 data. The experimental analysis yielded a 99.46 % accuracy rate, a 99.41 % recall rate, and a 99.51 % precision rate.

Although there have been numerous researches on the design of IDS for vehicular networks, the most of them focus solely on detecting known attacks within the vehicle itself or on external networks. In addition, a limited number of articles [19,20] focus only on a certain sort of attack, such as DoS or spoofing attacks. Nevertheless, in practical scenarios, both the internal networks within vehicles and the external networks they connect to are susceptible to a wide range of attacks, including both known and emerging zero-day attack patterns. The IDS proposed in Refs. [25,28,29] are the only research studies that take into account both the CAN bus and external networks. Additionally, the IDS proposed in Refs. [28,30], and [31] are the only techniques capable of detecting both unknown and known attacks. Therefore, it is necessary to develop an IDS that can identify both previously unknown attacks (zero-day attacks) and known attacks on both the internal vehicle network and the external vehicular networks. The objective of our propose IDS is to accomplish this goal. However, in order to ensure the reliability and practicality of IDSs in real-world vehicle systems, it is crucial to assess the comprehensibility and interpretability of the model. Unfortunately, previous research has not addressed this aspect, leaving the feasibility of other techniques in real-world IoVs unproven. Therefore, we have evaluated our proposed IDS using the SHapley Additive exPlanations mechanism from Explainable AI. This evaluation enhances the trustworthiness and interpretability of our IDS, ultimately increasing user confidence and driver safety. Table 1 presents the comparative analysis of the existing models.

2.1. Intra-vehicles network vulnerabilities

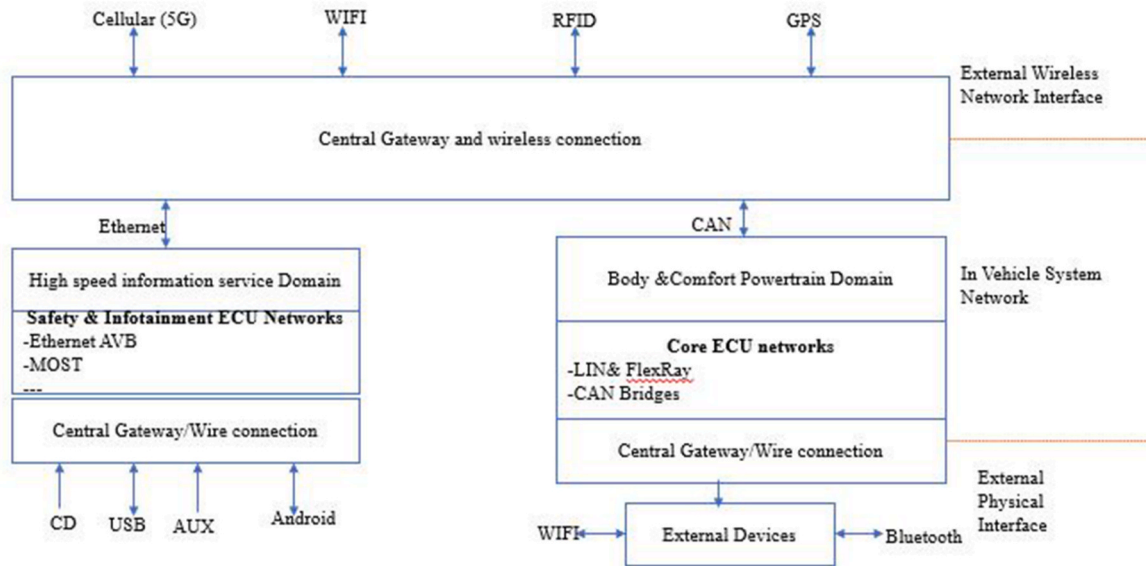
The in-vehicle electronic system consists of several ECUs that are connected through various communication methods, including IVNs such as local interconnect network, media-oriented system transport (MOST), FlexRay, and CAN bridge [33,34]. External networks are connected to the central gateways for communication. Gateways specifically allow for communication within a vehicle's network, enabling a variety of functions [35]. Fig. 2 depicts the intricate arrangement of connections within in-vehicle systems, as well as their connection to the outside world.

The services offered encompass gas and toll payment, fleet administration, automatic collision notification, vehicle diagnostics, safety services, and entertaining. The on-board diagnostics 2 interface is utilized for the purpose of exchanging information and status between ECUs. Moreover, the implementation of V2V and V2I communication systems enables the enhancement of safety for automobiles, drivers, occupants, and pedestrians [36]. Fig. 2 illustrates the several external interfaces of the IVN, which consist of the IVN itself, an external physical network interface, and an external interface of wireless network. Malicious hackers have the ability to carry out various forms of assaults against IoVs through In-Vehicle Networks due to their inadequate

Table 1

Summary of the existing state-of-the-art models.

Refs/Algorithms	External Network Attack Detection	Intra-vehicle Network Attack Detection	Zero-Day Scenario	Botnet Attack Detection	XAI	Results
[14]LeNet	×	✓	×	×	×	CANBus: Accuracy = 98.10 %, recall = 98.04 %, precision = 98.14 %, F1 = 97.83
[19]DL	×	✓	×	×	×	CAN Bus: Accuracy = 98 %
[20]KNN, and SVM	×	✓	×	×	×	KNN: Accuracy = 98.3 %, Recall = 96.9 %, Precision = 97.4 %, and F1 = 98.2 %
DCNN [22]	×	✓	×	×	×	Accuracy = 99.93 %; Recall = 99.84 %; F1-score = 99.91 %.
[10]GAN	×	✓	×	×	×	CAN bus: Accuracy = 97.9 %, Precision = 96.8 %, AUC = 99 %, and DR = 99.6 %
[23] RNN-LSTM	×	✓	×	×	×	Accuracy = 98.76
[24]NN-LSTM	×	✓	×	×	×	Accuracy = 99.99 %, recall = 99.94 %, F1 = 99.97 %, and AUC = 100 %.
DLSTM-AE [25]	✓	✓	×	×	×	UNSW-NB15: Precision = 100 %, recall = 97.0 %, F1-score = 98.0 %, and the accuracy = 96.0 %
[28] MTHIDS	✓	✓	✓	✓	×	CANbus: DR = 99.99, Accuracy = 99.99; F1 = 99.99. CICIDS: DR = 99.81 %, Accuracy = 99.87 %; F1 = 99.87 %
[29]TL	✓	✓	×	✓	×	CANbus: Recall = 100 %, Accuracy = 100 %; F1 = 100 %, Precision = 100 %: CICIDS: Recall = 99.92 %, Accuracy = 99.92 %, F1 = 99.92 %, Precision = 99.92 %
BPNN [26]	✓	×	✓	×	×	Kyoto dataset: Recall = 99.06 %; Accuracy = 99.05 %.
MLP [27]	✓	×	×	×	×	CICIDS2017: Accuracy = 99.46 %, recall = 99.41 %, and Precision = 99.51 %.
KNN and CART	✓	×	×	×	×	KNN: Accuracy = 99.79 %, CART Accuracy = 99.97 %.
[32]DBN	✓	×	×	×	×	DR = 99.92 %, Accuracy = 99.43 %, F1 = 96.0 %
Ours	✓	✓	✓	✓	✓	

**Fig. 2.** Interface types of intra-vehicle networks.

security measures. These measures include the lack of encryption mechanisms, and authentication as well as the presence of numerous interfaces that are both big in number and continuously growing [37]. The cyber-attacks can be initiated utilizing any of the interfaces, specifically a physical, cable, or wireless interface.

2.2. Materials and methods

2.3. Proposed XAI ensemble TL IDS architecture

The objective of this research is to develop an IDS capable of identifying zero-day botnets and different forms of attacks in both intra-vehicle and external vehicular networks, with the aim of safeguarding them. Fig. 3 illustrates the standard attack scenario and the structure of a vehicle protected by an IDS.

Adversaries have the ability to initiate internal assaults on In-vehicle networks by exploiting the On-board diagnostics II interface [38]. They can also start external attacks on external vehicular networks by transmitting malevolent traffic packets over wireless interfaces. Therefore, it is recommended to implement the proposed IDS in both In-vehicle networks and external networks. The proposed IDS can be implemented on the CAN bus in IVNs and integrated into the gateways to identify and raise alerts for aberrant CAN signals and prevent any malicious traffic that attempt to compromise IoV [13]. Fig. 3 provides a synopsis of the proposed IDS model. While Fig. 4 showed a snapshot of the whole XAIEnsembleTL-IoV process.

2.4. Datasets description and transformation

The evaluation of the developed IDS utilizes two datasets: 1) the vehicle hacking dataset [10] for assessing the detection of abnormalities in in-vehicle communication, and 2) the N-BaIoT data for evaluating the

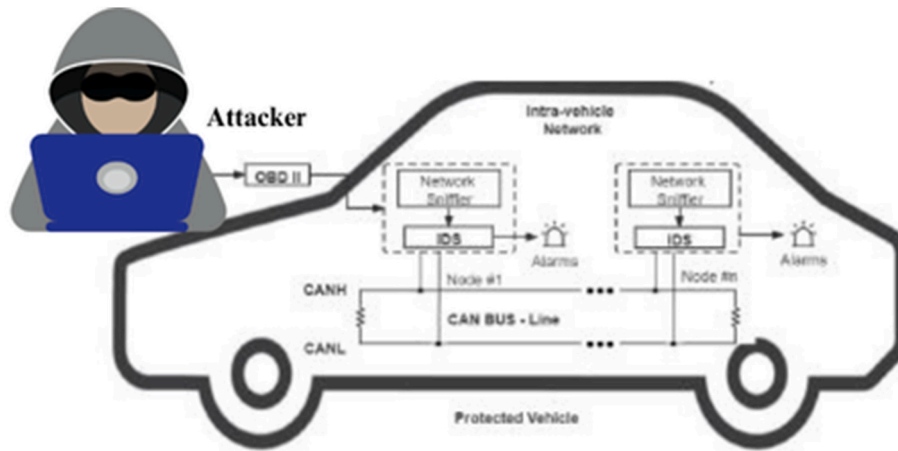


Fig. 3. The Internet of vehicle architecture protected by IDS.

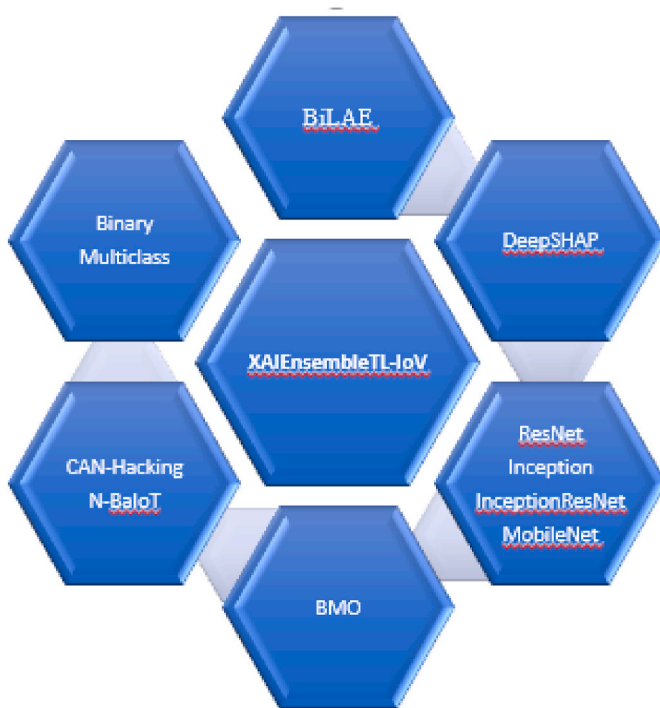


Fig. 4. XAIEnsembleTL-IoV process.

detection of anomalies in external networks. The two data are described as follows:

2.4.1. The vehicle hacking data

The data was generated by recording logs of CAN traffic through the On-Board diagnostics (OBD-II) port of an actual vehicle [10]. The logs were acquired during the ongoing attacks. The dataset encompasses many attack types, such as fuzzy attacks, DoS attacks, RPM gauge spoofing, and drive gear spoofing. Each dataset consists of a duration of 30–40 min of CAN traffic. There was a total of 300 instances of message injection, with each attack lasting for a duration of 3–5 s. Table 2 provides comprehensive information regarding the vehicle hacking dataset. The CAN dataset is available at <https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset>.

2.4.2. The N-BaIoT dataset

The N-BaIoT data set [39] is available to the public for cyber security research purposes. It consists of network traffic data generated by an IoT

Table 2
Size and label of the CAN-Data.

Attack/Label	Original Number of Samples	Test Sample Number	Training Set Sample
Normal	14,037,293	4,211,188	9,826,105
Fuzzy attack	491,847	147,554	344,293
DoS attack	587,521	176,256	411,265
RPM gauge spoofing	654,897	196,469	458,428
Gear spoofing	597,252	179,176	418,076

testbed, which included two doorbells, a baby monitor, a thermostat, a webcam, and four security cameras. These commercial IoT devices were infected with BASHLITE botnets and Mirai. For this research, we used 363,979 samples of benign IoT network traffic and 1,483,658 samples of IoT botnet traffic [40]. The network traffic features were normalized using a specific method to ensure no bias towards any particular feature. The N-BIoT data is utilized for three distinct classification tasks as shown in Table 3: binary classification, 3-class classification, and 9-class classification. Binary categorization distinguishes between harmful traffic and benign traffic. The 3-class system offers a more thorough examination of malware by categorizing data points into three specific categories: benign, mirai, and gafgyt. In the 9-class classification, the data points have been categorized into various attack types: benign, ack,

Table 3
Description of the N-BaIoT dataset.

Scenario	Classes	Class Name
Binary	Attack	Benign
	Benign	Genuine network data
3-class	Mirai	Network traffic contaminated with malware
	Gafgyt	Network traffic contaminated with malware
	Benign	Genuine network data
	Benign	Genuine network data
	ACK	Malicious software transmitting unsolicited data
	Junk	
	Combo	Malware is transmitting unsolicited data and establishing a connection to a certain port and IP address.
9-class	SYN	Malware that utilizes SYN-flooding techniques
	Scan	Perform a comprehensive scan of the network devices to identify any potential vulnerabilities.
	User Datagram Protocol	User datagram protocol flooding (Gafgyt & Mirai)
	TCP	Malware transmission control protocol flooding
	UPDPlain	Malware is being used to flood UDP packets at a lower rate, reducing the possibility of sending a bigger number of packets per second.

compact, scan, junk, syn, udp, tcp, and udplain. The data is available at <https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot>.

There is a lack of publicly available benchmark datasets for the development of external network IDS due to challenges related to commercialization, popularization, and privacy concerns [7,41]. Wireless local area networks and cellular networks are frequently used for communication in the IoV and connected automobiles. Therefore, assaults on traditional computer networks might be seen as analogous to intrusions on external vehicular networks. Consequently, numerous research projects and papers [32,42] have created external vehicular networks by utilizing comprehensive network security data such as KDD-99, CICIDS 2017, AWS, NSL-KDD, BoT-IoT, UNSW-NB15, and N-BaIoT [43]. Out of all the cyber-security datasets, N-BaIoT stands out as the most comprehensive dataset for contemporary external networks. It is considered state-of-the-art and includes a greater number of features, occurrences, and types of cyber-attacks compared to other datasets [27].

2.5. Preparation of the dataset

Prior to designing the XAI ensemble transfer learning, it is necessary to clean and prepare the network traffic data in order to attain best performance and enhance the learning procedure. Three essential processes are utilized during the process of data preparation. The initial stage involves data cleansing, followed by data manipulation, and culminates with data labelling and annotation.

2.5.1. Data cleansing

Due to its substantial magnitude, this dataset is highly susceptible to missing and noisy data. The CAN dataset has a grand total of 1,270,310 cases, which includes both noisy and inconsistent data. This subsection covers the fundamental stages involved in data preprocessing. Initially, we employed a range of methodologies to eliminate unwanted disturbances and rectify irregularities in the dataset. Specifically, we utilized interquartile range to identify and address outliers, and employed the predictive mean matching to fill in any missing values. Subsequently, to implement the chosen models, we transformed the qualitative values into numeric values. However, the dataset includes a feature called Data_Field, which consists of eight-byte hexadecimal numbers separated by spaces. The space between bytes has been eliminated by utilizing the gsub function. Subsequently, the hexadecimal values have been transformed into decimal values using the Rmpfr function, as the majority of the data field exceeds 64 bits. In addition, many TL models have undergone a reshaping of the input data shape into three dimensions. This was achieved by utilizing numpy's reshape function together with the concatenate methods and swapaxes.

2.5.2. Data manipulation

We have implemented this measure in order to enhance productivity and gain a deeper comprehension of the patterns. Certain features exhibit greater magnitudes, resulting in inaccurate performance. Equation (1) demonstrates the use of a minimum-maximum normalization to rescale the selected feature values within a specific range.

$$Y_{\text{norm}} = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \quad (1)$$

where y is a feature vector of network traffic, with $y_{\min}$ being the minimum value and $y_{\max}$ being the highest value of y . This technique preserves the original arrangement of network traffic attributes.

2.5.3. Data labelling and annotation

Once the data has been obtained, it must undergo pre-processing in order to create a suitable input for the proposed IDS. Following the process of data normalization and manipulation, the data samples are

transformed into segments according to the feature sizes and timestamps of network traffic datasets. During the data annotation step, the converted data is assigned labels according to the attack patterns found in the data chunks. If every sample within a chunk is classified as normal, the corresponding data is labeled as "normal".

2.6. Bidirectional Long Short-Term Memory autoencoder (BiLAE)

An autoencoder is a DL method that uses unsupervised learning to assess the dynamic correlations between the features of high-dimensional data and generate a lower-dimensional representation in a latent space [44]. BiLAE, in contrast to the traditional Autoencoder, utilizes BiLSTM units to capture the long-term interconnected variations in network traffic characteristics. The BiLAE method is introduced in this subsection to decrease the dimensionality of IoV network traffic data and provide a low-dimensional latent space feature representation with minimal reconstruction error. The representation of a sequence of network traffic attributes is done using a three-dimensional matrix, $Z \in \mathbb{R}^{k \times 1 \times n}$, as shown in Equation (2):

$$Z = \{Z_{f,i}\}_{i=1}^k \quad (2)$$

The equation $\{Z_f\}_{f=1}^k = Z_1, Z_2, \dots, Z_k$, $n \in X^+$ represents the relationship between network traffic features and instances in a dataset. Here, $k \in X^+$, represents the number of network traffic characteristics, and represents the total examples of network traffic accessible in the data. Equation (3) represents a single occurrence of network traffic.

$$\{z_i\}_{i=1}^n = z_1, z_2, \dots, z_n \quad (3)$$

The feature vector $z_i \in \mathbb{R}^k$ belongs to the set of real numbers, $\mathbb{R}$, and represents a network traffic feature. The encoder component of a single hidden layered BiLSTM autoencoder [45] was employed to compress the network traffic attribute matrix, as defined by Equation (3), in order to decrease its dimensionality while preserving the information inherent in the original data. The BiLSTM model possesses the potential to acquire extensive temporal dependencies through its feedback interconnections and a recurrent memory unit that is regulated by three gates: the input gate, the forget gate, and the output gate [46]. BiLAE is capable of processing high-dimensional network traffic feature set, Z , and generating a low-dimensional latent-space representation, $\tilde{Z}$, in the hidden layer. The input gate (in_i) vector, forget gate (fn_i) vector, memory cell state (cn_i) vector, output gate (on_i) vector, and hidden state (hn_i) vector were created using the BiLAE. The dimension of the column vectors is represented by $in_i, fn_i, cn_i, on_i, hn_i \in \mathbb{R}^{du}$ where du is the number of BiLSTM hidden units that provide the required network traffic feature dimensionality.

The weight matrices and bias vectors were acquired through training the BiLAE utilizing the back propagation through time technique [47]. The weight matrix structures for the links between the input and the recurrent gates are denoted as $W_{inz}, W_{fnz}, W_{cnz}, W_{onz} \in \mathbb{R}^{du \times k}$. These matrices represent the connections between the input and the input gate, forget gate, cell, and output gate, respectively. The weight matrices for connections among the recurrent gates and the hidden state are denoted as $W_{inhd}, W_{fnhd}, W_{cnhd}, W_{onhd} \in \mathbb{R}^{du \times du}$. These matrices represent the connections from the input gate to the hidden state, the forget gate to the hidden state, the cell state to the hidden state, and the output gate to the hidden state, respectively. The bias vectors for the input gate, forget gate, cell state, and output gate are $b_{in}, b_{fn}, b_{cn}, b_{on} \in \mathbb{R}^{du}$, respectively. The recurrent function of activation is a sigmoid function denoted by σ_{rec} . The activation of the hidden layer is denoted by the function σ_{hd} . Diagonal matrices, M_{in} and M_{fn} , are constructed using the input gate and forget gate vectors, as denoted by Equations (4) and (5), respectively.

$$M_{in} = \text{diag}(in) = [in_1 \dots in_{nd}] \quad (4)$$

$$M_{fn} = \text{diag}(fn) = [f_{n1} \dots f_{nnd}], \quad (5)$$

For z_i , the output encoded is shown in Equation (6).

$$\tilde{z}_i = n_\phi(z_i, cn_{i-1}), \quad (6)$$

Where cn_{i-1} is the previously cell vector state and n_ϕ is the function encoding. Equation (7) represents the low-dimensional traffic feature matrix.

$$\tilde{Z} = \left\{ \left\{ \tilde{z}_{f,i} \right\}_{i=1}^d \right\}_{f=1}^k \quad (7)$$

2.7. Bidirectional Long Short-Term Memory (BiLSTM)

The conventional LSTM model is limited to capturing the relationship between the present state and the previous context in a unidirectional manner [48,49]. BiLSTM, on the other hand, utilizes two LSTM hidden layers to examine input data sequence in both positive and negative time directions, allowing it to have complete access to both past and future sequential information [50]. Hence, a sophisticated BiLSTM is devised to effectively identify IoV botnet attack traffic by examining the prolonged interconnected variations in low-dimensional characteristics generated by BiLAE. The BiLSTM model was used to process the reduced IoV network traffic feature set, $\tilde{Z}$, and its related target vector, y' , resulting in the generation of input gate ($\vec{i}_{ni}, \overleftarrow{i}_{ni}$) vectors, forget gate ($\vec{f}_{ni}, \overleftarrow{f}_{ni}$) vector, memory cell ($\vec{c}_{ni}, \overleftarrow{c}_{ni}$) state vector, hidden state ($\vec{h}_{ni}, \overleftarrow{h}_{ni}$) vector, and output

($\vec{o}_{ni}, \overleftarrow{o}_{ni}$) state vector. The column vectors are denoted as $\vec{i}_{ni}, \overleftarrow{i}_{ni}, \vec{f}_{ni}, \overleftarrow{f}_{ni}, \vec{c}_{ni}, \overleftarrow{c}_{ni}, \vec{o}_{ni}, \overleftarrow{o}_{ni}, \vec{h}_{ni}, \overleftarrow{h}_{ni}$. The weight matrices ($\vec{W}_{g(\cdot)}, \overleftarrow{W}_{g(\cdot)}$) and bias vectors ($\vec{b}_{in(\cdot)}, \overleftarrow{b}_{in(\cdot)}$) were acquired through training the BiLSTM using the BPTT algorithm. The recurrent activation function is a sigmoid function, denoted as σ_r . The activation function for the hidden layer is denoted as $\sigma_{h_{ni}}$. The activation function for the output layer is a softmax function denoted as σ_y . The forward BiLSTM hidden layer's parameters are determined based on the input data sequence from the past to the present, while the backward BiLSTM hidden layer's parameters are calculated based on the input data sequence from the future to the present. Equation (8) minimizes the discrepancy between the projected output of BiLAE, y' , and the target output, y .

$$\theta(y, y') = \left[\theta(y_f, y'_f) \right]_{f=1}^k \quad (8)$$

The variable θ can be a binary cross-entropy loss function, used for binary classification, or categorical cross-entropy loss function, used for multi-class classification.

2.8. Transfer learning

Transfer Learning (TL) refers to the practice of transferring the learned weights of a neural network model from one data to a different data [51]. The TL technique has been effectively utilized in numerous image processing jobs. The reason for this is that the lower layers of CNN models typically learn broad patterns that may be applied to various tasks, whereas the higher levels learn unique features that are tailored to specific data sets. Hence, the lower levels of CNN models can be directly translated to various jobs. In order to enhance the efficiency of TL, the technique of fine-tuning can be employed throughout the TL process of DL models. Fine-tuning allows the learning model to update the more complex characteristics in the pretrained model in order to better adapt to the target task [51]. Within the proposed architecture, we have chosen ResNet, Inception ResNet, Inception, and MobileNet as the

foundational CNN models. These models have demonstrated significant achievements in the majority of image classification tasks [52]. The choice of these CNN architectures and optimization techniques was made based on their proven effectiveness in image recognition tasks and transfer learning applications. Each architecture's performance was assessed for its suitability in detecting zero-day botnet attacks in vehicular networks. Table 4 highlights the roles and advantages of each of the CNN-TL architectures used in this research.

2.9. Explainable A.I mechanism

Shapley Additive Explanations (SHAPley) is a game-theoretic technique employed to offer explanations for the output of any ML model. The method optimizes the allocation of credit by utilizing the widely recognized Shapley values from game theory, together with their corresponding expansions, and provides localized explanations [53]. The CNN-transfer learning has been increasingly popular in recent years due to its exceptional performance. However, they are considered to be opaque entities. On the other hand, XAI clarifies and interprets the decision made by TL. In this research, we utilized the deep SHaPley XAI technique to provide explanations for the predictions generated by our proposed ensemble CNN-TL.

2.9.1. Deep SHaPley method

Deep SHaPley is a rapid and model-specific method used to explain the predictions made by CNN [53]. This approach was derived from the established correlation between Shapley values and DeepLIFT. Deep SHaP leverages the back-propagation technique used by DeepLIFT to efficiently calculate approximate Shapley values for a CNN. DeepLIFT's multiplier is formally defined as the ratio of the influence of Δ_x to Δ_t separate by Δ_x .

$$M\Delta_x\Delta_t = \frac{C\Delta_x\Delta_t}{\Delta_x} \quad (9)$$

Where $\Delta_x = x-r$ and r are the input reference.

2.9.2. Proposed framework of the XAI ensemble TL model

This section presents a framework for elucidating the predictions made by ensemble TL-based IDS in IoV. The framework utilizes the deep SHAPley algorithm, which was previously covered in Section 4.4.1. The objective of the proposed framework is to offer a comprehensive understanding of how choices made by ensemble TL-based NIDS can be elucidated to security analysts. These professionals depend on this decision to safeguard the security and safety of IoV and develop more cyber-resilient IoV. The suggested framework is capable of identifying and elucidating various types of cyber-attacks, including Gear Spoofing,

Table 4
Architectures of the CNN-TL models.

CNN-TL Architectures	Roles/Merits
ResNet	A deep convolutional network that utilizes residual connections to allow the training of very deep networks without vanishing gradients. Its ability to maintain performance as depth increases made it a natural choice for our model.
InceptionResNet	This architecture combines the strengths of both Inception and ResNet, using residual connections alongside the Inception structure for improved efficiency and accuracy.
Inception	Known for its multi-scale architecture, Inception employs parallel convolutional filters of different sizes, which allows the network to capture different features at various levels of abstraction.
MobileNet	A lightweight model optimized for mobile and resource-constrained devices. We included it to evaluate the trade-off between performance and computational efficiency in detecting botnet attacks in vehicular networks.

DoS, Fuzzy, RPM spoofing, Mirai, and Bashlite, within IoV networks. The system depicted in Fig. 5 relies on generative deep learning, which acquires knowledge from time-series data obtained from IoV networks. It is capable of discerning between legal and questionable cases by analyzing the most significant aspects. The framework consists of three primary components: a NIDS based on Transfer learning, Explanation module, and BiLAE module. The uniqueness of this research lies in the utilization of the explanation module, which employs the deep SHAPley framework to offer global explanations for the detection produced by the IDS in IoV networks. Compared to other frameworks designed for privacy and security in IoV [54], our model includes an explainable module that enhances the transparency of any IDS. Moreover, the knowledge obtained from the explanations at the global level can be utilized to advance the capability of IoV systems to endure and recover from cyber-attacks.

2.9.3. Global explanation technique

The SHAP framework offers the advantage of providing a comprehensive insightful explanation and interpretation on a global scale. It offers a comprehensive summary of the key qualities that are crucial for a model. The authors [53] proposed a method for comprehensively describing a model by integrating ShaPley values in a matrix format.

This matrix includes one row for each data instance and one column for each feature [55]. Within the framework of our TL based NIDS, the SHAPley global explanation method can be employed to identify the precise features in the CAN-hacking and N-BaIoT that either enhance or diminish the probability of a cyber-attack. Security specialists can utilize this knowledge to create IDS that are impervious to cyber threats in IoV.

2.10. The ensemble learning strategy proposed

Ensemble learning (EL) is a method that combines numerous base learning models to create an EL model that has enhanced performance. EL is commonly employed in data analytics challenges because of the superior performance of several learners working together compared to individual learners [28]. In this research, we used soft voting ensemble (probability voting) in which each CNN-TL model predicts the probability of each class label, and the final prediction is the class with the highest average probability. This method takes into account the confidence of each CNN-TL model's prediction. In soft voting, each classifier provides a probability $P_i(z|y)$ for each class z . The final prediction is the class with the highest average probability. Let $P_i(z_k|y)$ be the probability that classifier C_i assigns to class z_k . The soft voting method can be represented as:

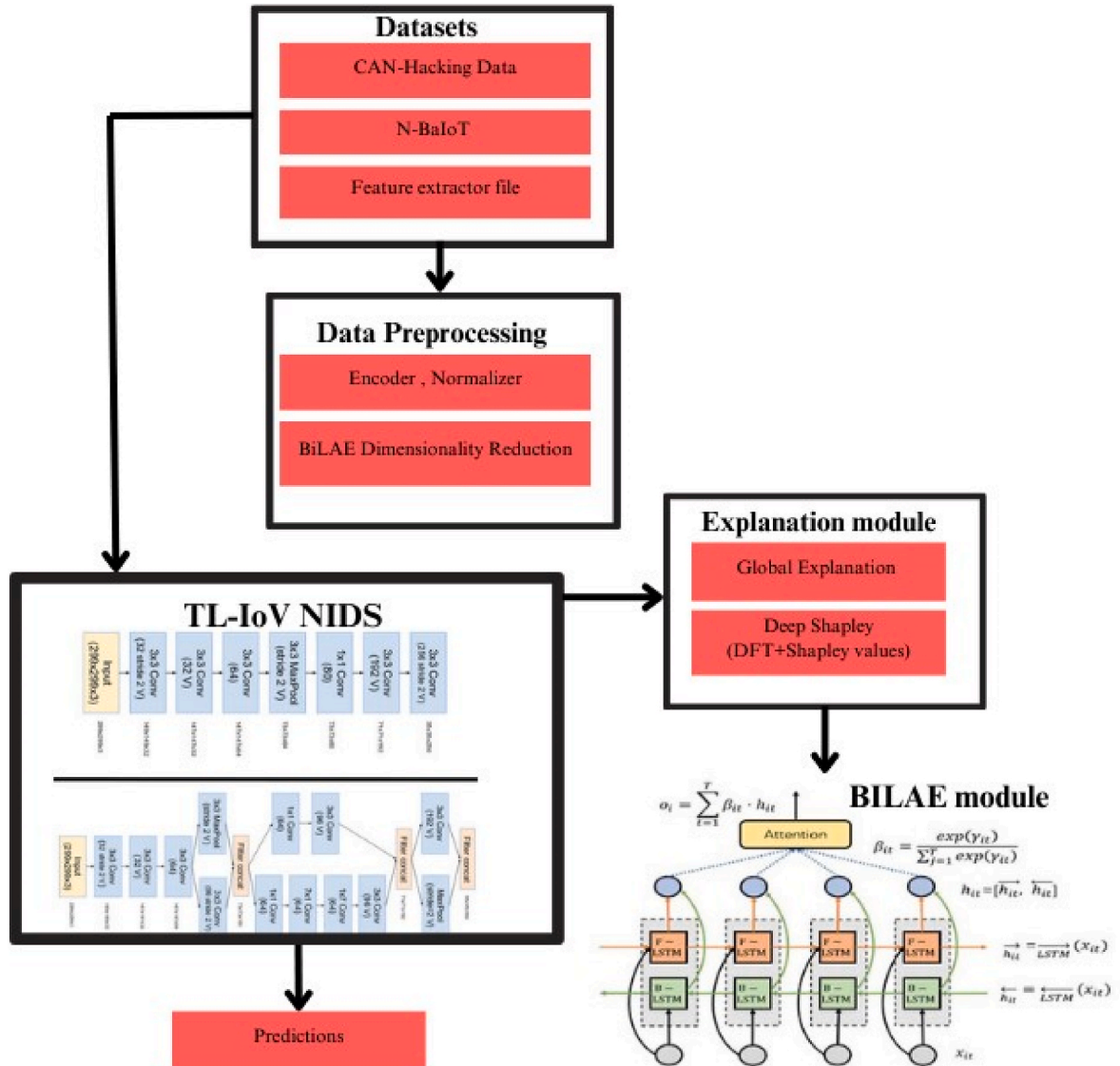


Fig. 5. Proposed XAI Ensemble TL framework for zero-day botnet detection in IoV.

$$P(z_k|y) = \frac{1}{N} \sum_{i=1}^N P_i(z_k|y) \quad (10)$$

Therefore, the final prediction is:

$$Z = \operatorname{argmax}_{z_k} P(z_k|y) \quad (11)$$

This involved calculating the mean probability derived from four (4) CNN-TL architectures for detecting intrusions in the IoV using the Car hacking and N-BaIoT data. Below is a detailed breakdown of the process we used to accomplish this:

Stage 1: Data Preprocessing: We do preprocess on the datasets (Car hacking and N-BaIoT) by partitioning them into distinct training and testing subsets. Every subset consists of the target labels (benign and malicious) and the corresponding features for the specific dataset.

Stage 2: Individual Learning Models: Choose a set of autonomous CNN-TL architecture, specifically ResNet, Inception ResNet, Inception, Mobile Net, that we plan to merge into an ensemble.

Stage 3: Train the individual TL models: For each individual that is chosen, the CNN-TL architecture used will include ResNet, InceptionResNet, Inception, and Mobile Net. The algorithms were trained using training data from both the (Car hacking and N-BaIoT) datasets. We obtained a collection of trained models, each with the ability to make predictions for intrusion detection.

Stage 4: Prediction Probability: For every model that has been trained, we utilize it to provide predictions on our testing dataset. Instead of solely acquiring the ultimate prediction label, our focus lies in the projected probabilities of intrusion (namely, label attack) for each individual instance.

Stage 5: Voting Ensemble: We computed the mean of the projected probability from each individual model for every case in our testing dataset. The mean can be calculated for class 1 (attack).

Stage 6: Evaluation of the Ensemble TL model: We assessed the effectiveness of our voting ensemble TL models by measuring standard measures, including accuracy, recall, precision, F1-score, and MCC. In addition, we evaluate the efficacy of the ensemble TL by comparing these findings with the performance of individual CNN-TL models.

2.11. Hyperparameter optimization of the ensemble CNN-TL architectures with barnacles mating optimizer

In order to enhance the performance of the ResNet, InceptionResNet, Inception, and Mobile Net CNN-TL architecture and better adapt them to the selected data, it is necessary to tune and optimize the hyper-parameters. Like other DL models, TL models have a significant number of hyper-parameters that require tuning. In the proposed TL framework, the model-design hyper-parameters was employed which involve the percentage of frozen layers, the dropout rate, and the learning rate. Hyperparameter optimization (HPO) refers to the automatic optimization of hyper-parameters in ML or DL models utilizing optimization techniques [56]. This study utilizes the Barnacle Mating Optimizer (BMO) algorithm, which is one of the optimization strategies employed for HPO problems. The authors [57] introduced a novel evolutionary algorithm known as the BMO approach. This algorithm draws inspiration from the reproductive behavior of barnacles, namely their ability to reproduce through both female and male mechanisms. The solution in BMO is represented by a symbol that bears resemblance to a chromosome in GA and a particle in PSO. Barnacles reproduce by the process of fertilization, which involves the mating of sperm and natural intercourse. By engaging in random locomotion, they actively seek out a mate and subsequently deposit their sperm into the hollow of the partner's mantle. Conversely, when the sperm is discharged into the seawater to fertilize the barnacle eggs, the process of sperm mating occurs. Fig. 6 depicts the life cycle of barnacles. Barnacles undergo a larval growth process consisting of a non-feeding pre-settling cyprid

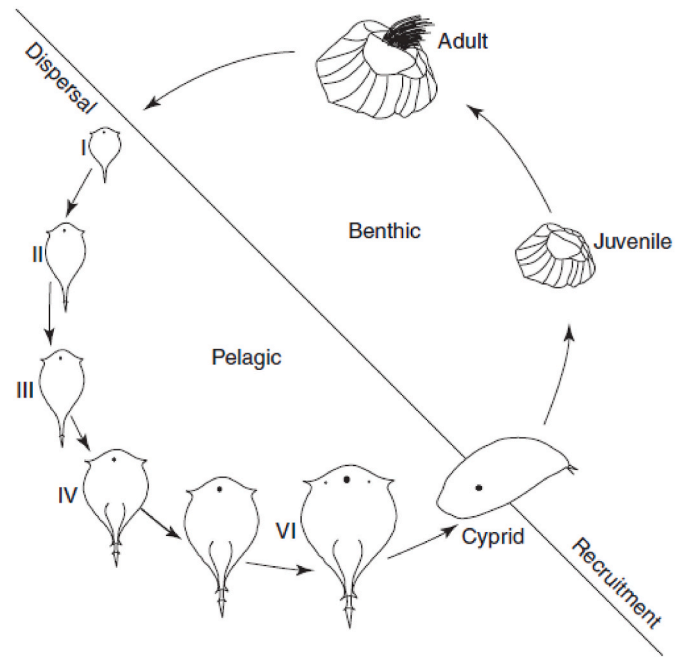


Fig. 6. The life cycle that barnacles undergo [58].

instar and six naupliar instars [58]. Once the cyprid has attached itself to a surface, it undergoes metamorphosis and transforms into an adult.

The BMA optimization approach consists of three primary steps.

1. **Initialization Phase:** Like other optimization methods, the initialization steps in BMO involve generating the initial population randomly. The initial population is represented as a matrix of decision parameters in the following manner:

$$Y = \begin{bmatrix} y_1^1 & y_1^2 & y_1^N \\ y_2^1 & y_2^2 & y_2^N \\ \vdots & \vdots & \vdots \\ y_n^1 & y_n^2 & y_n^N \end{bmatrix} \quad (12)$$

N represents the quantity of control variables, whereas n represents either the population size or the number of barnacles. The control variables in Equation (12) are constrained by the lower and upper bounds of the problem, as follows:

$$up = [up_1, \dots, up_i] \quad (13)$$

$$lo = [lo_1, \dots, lo_i] \quad (14)$$

The equation uses up and lo to represent the upper and lower boundaries of the i th parameter. In this context, each barnacle in the initial population is evaluated using an objective function. Subsequently, a sorting approach is employed to place the optimal solution at the top of the solution matrix.

2. **Choice Process Phase:** The selecting procedure of two barnacles can be done randomly based on the length of their penises, p_{ig} . The selection process was a random procedure limited to the measurement of barnacle penis length. Each barnacle exchanges its sperm with other individuals. However, each individual barnacle would undergo fertilization just once by a single person.

During a particular repetition, when the option exceeds the pre-determined p_{ig} , the process of sperm casting will occur. The factor described here elucidates that the BMO approach encompasses both exploration and exploitation stages. The process of sperm casting involves the ongoing generation of children. This word refers to the exploration and selection of sperm, which is done randomly. The

mathematical formula below demonstrates that the choice was made randomly, after which the initial assumption was fulfilled.

$$\text{bar_d} = \text{ranperm}(n) \quad (15)$$

$$\text{bar_m} = \text{ranperm}(n) \quad (16)$$

The parents to be married are bar_d and bar_m , and the number of populations is n .

Equations (15) and (16) demonstrate that the selection process is conducted in a random manner and satisfies assumption number 1 as stated in the preceding sub section.

3.Reproduction Process Phase: However, no analytical method was implemented for the reproduction of barnacles. The BMO approach specifically emphasizes the characteristics of genotype or the inheritance frequencies of parents in generating offspring, based on the concept of Hardy-Weinberg. Each new offspring is obtained in the following manner:

$$y_i^{N_{\text{new}}} = p y_{\text{bar_d}}^N + q y_{\text{bar_m}}^N \quad (17)$$

In Equation (17), P represents a pseudo-random usually distributed with $[0, 1]$, $q = (1 - p)$, and $y_{\text{bar_d}}^N$ and $y_{\text{bar_m}}^N$ denotes the parameter of individual dad and mum correspondingly. Sperm casting occurs when the selection of barnacles to mate with above the threshold value of p_{lg} , which is initially established as follows.

$$Y_i^{N_{\text{new}}} = \text{rand} * Y_{\text{bar_m}}^n \quad (18)$$

Equation (18) uses the symbol rand to represent a random integer between 0 and 1. The exploration process generates new offspring using the individual Mum. The BMO algorithm in this research calculates a fitness function that aims to minimize the classification error rate. The fitness function obtained from the BMO algorithm is calculated in the following manner:

$$\text{Fitness function} = \frac{\text{Number of missclassified samples}}{\text{Total number of samples}} * 100 \quad (19)$$

BMO role in the research is to optimize the hyperparameters of the model, helping to achieve higher performance in terms of detection accuracy. BMO was selected for its ability to efficiently explore complex optimization spaces and its proven success in other cybersecurity and machine learning applications.

3. Discussion and experimental setup

The experiments were carried out with the Scikit-learn and Keras packages in the Python PL. The analysis on the two datasets, which consisted of both the training and test data, was conducted using a Python script. Initially, we utilized the `train_test_split` technique from the scikit-learn library to divide both the CAN-hacking and N-BaIoT datasets into training (80 %) and test (20 %) datasets. The training dataset was utilized to train the chosen models, while the test dataset was employed to further evaluate the learned ensemble TL. In addition, we further divide the training data into two parts: 80 % for training the selected model and 20 % for optimizing hyperparameters via Barnacle mating optimization. The training dataset for CAN-Hacking and N-BaIoT has been allocated 80 % of the total data, while the remaining 20 % has been set aside for the test dataset, as recommended in Ref. [59]. In order to mitigate the issue of over-fitting, researchers have determined that a specific splitting ratio between the training and test datasets, as described in Ref. [59], is the most effective. Ultimately, a range of performance measures has been employed to assess the overall success of the suggested models. In order to assess the performance of our research model in relation to the Ground Truth equations (20)–(23), we employ terminology such as True Negative (TN), False Positive (FP), False Negative (FN), and True Positive (TP).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (20)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (21)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (22)$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (23)$$

Where True-positive (TP) refers to the accurate identification of assault instances as attacks. A true-negative (TN) refers to the accurate identification of normal situations as normal. Where false positive (FP) refers to the count of typical events that are mistakenly identified as attacks. False-negative (FN) denotes the count of genuine assault occurrences that are erroneously identified as normal [60]. We have evaluated four CNN-TL architectures and the chosen architectures are Residual Network (ResNet), Inception, Inception ResNet, and mobileNet. The hyperparameters for all the selected models are as follows: the hidden layer function is ReLu, the batch size is 64, the loss function is categorical_crossentropy, the output layer activation function is softmax, and the optimizer is Adam. In order to create the most effective models, the key hyperparameters of all the CNN-TL architectures in the proposed framework were optimized using BMO. Table 5 displays the values of the hyperparameters of each of the CNN-TL architectures.

3.1. XAI explanations of the proposed ensemble TL model

This section describes how we use the Deep SHAPley explanation algorithm [61] to clarify the predictions generated by our proposed ensemble TL model. The Deep SHAPley algorithm is an efficient and model-specific approximation method that provides both local and global explanations for the outcomes generated by ensemble TL model. The creators of the SHAP algorithm have produced the Python SHAP package (<https://github.com/slundberg/shap>), which includes various visualization plots that allow for both local and global interpretation based on estimated SHAPley values. With the utilization of this Python module, we are able to elucidate the forecasts made by our proposed ensemble TL model.

In order to illustrate the most significant characteristics in our data that are crucial for the proposed binary and multi-class anomaly model to accurately forecast a specific attack category, we employ the SHAPley summary Plot. The SHAPley summary plots, unlike the force plots, provide a picturing plot that offers a global explanation of feature relevance for the model [62]. The most essential features are displayed at the top, while the least important characteristics are shown at the bottom. Fig. 7 displays a visual representation of the primary features that contribute to the multi-class IDS prediction of a DoS attack on the CAN-hacking data. In Fig. 8, we present a visualization of the primary attributes that contribute to the multi-class anomaly forecast of a Gafgyt

Table 5
Hyperparameters of the CNN-TL architectures.

Parameters	ResNet-	InceptionResNet	Inception	MobileNet
Units in Hidden Layer	128, 256, 128	32, 64, 32	21,806,882	31,545,670
Batch Size	64	64	64	64
Hidden activation layer	relu	linear	relu	relu
Dropout	0.1	NA	0.4	0.1
Output activation layer	softmax	softmax	Softmax	Softmax
Optimizer	Adam	Adam	Adam	Adam
Number of hidden layer	3	3	3	3

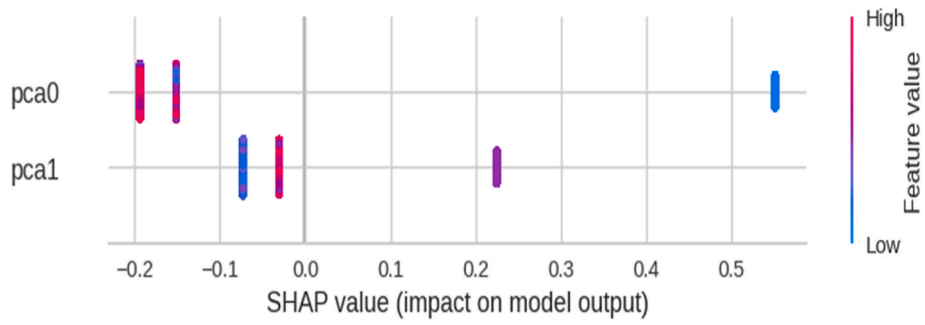


Fig. 7. Top attributes contributing to the detection of the DoS attack via the CAN-Hacking data.

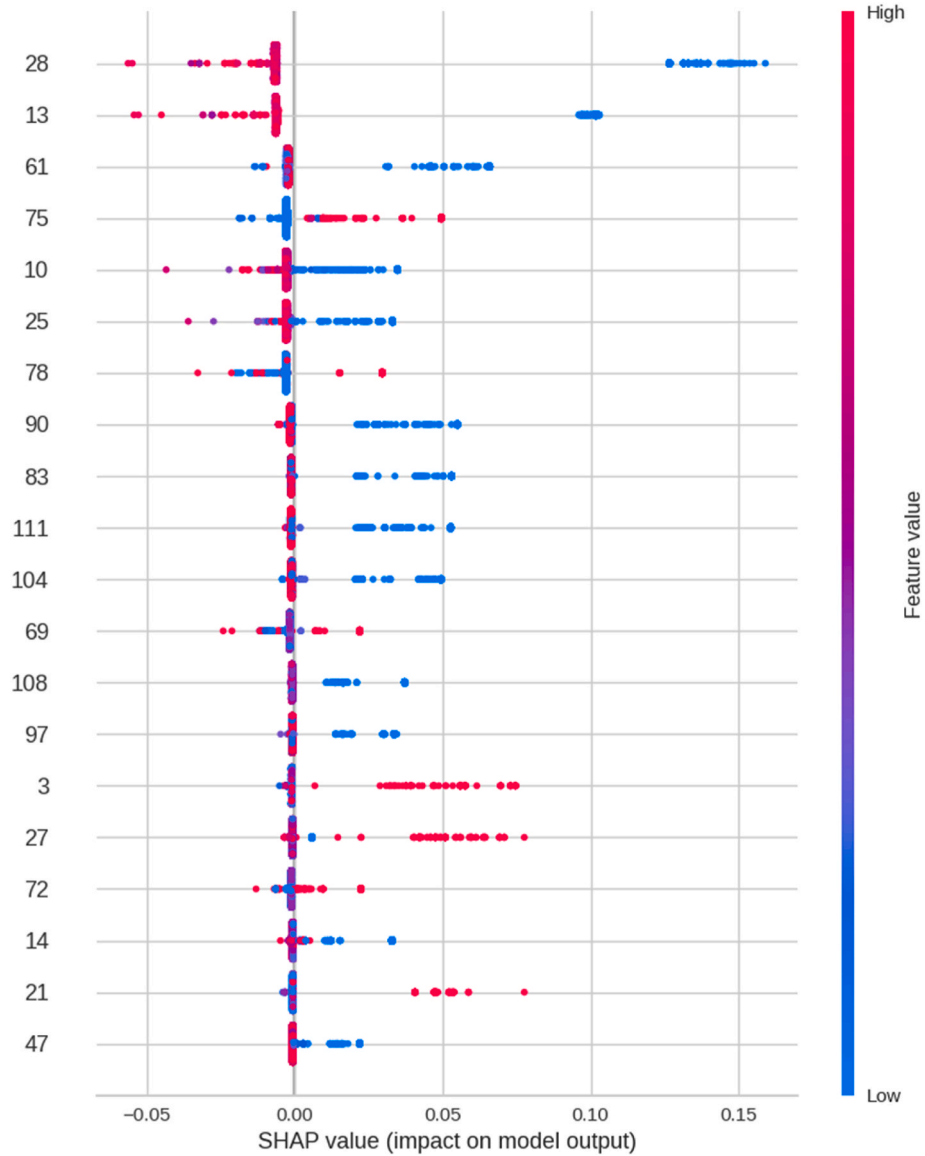


Fig. 8. Top attributes contributing to the detection of the Gafgyt attack in the N-BaIoT.

attack via the N-BaIoT.

3.2. Proposed ensemble TL performance analysis in binary scenario on intra-vehicle network traffic (Car-Hacking data)

Following BMO, each of the optimized TL models were utilized as

base learners to build the proposed ensemble TL models for Zero-day botnet detection in IoV. The performance of the optimized TL models and the proposed ensemble models on the Car-Hacking and N-BaIoT datasets can be found in Tables 6 and 8, respectively. Table 6 demonstrates that all optimized TL-based models, with the exception of ResNet and Inception, attain perfect accuracy, recall, F1-scores, precision, and

Table 6

Performance evaluation of optimized TL models and Ensemble TL models on Car-Hacking.

Method	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
ResNet-BMO	99.99	100	99.80	99.19	99.50	130.2	0.9
Inception ResNet-BMO	100	100	100	100	100	400.7	1.8
Inception-BMO	99.09	99.99	99.92	99.97	99.95	155.8	0.6
MobileNet-	100	100	100	100	100	595.8	2.2
EnsembleTL (Soft voting)	100	100	100	100	100	1050.8	2.9

MCC, all of which are 100 %. The proposed ensemble TL (soft voting) achieves perfect accuracy, recall, precision, F1-scores, and MCC, with a total training time of 1050. Thus, the suggested ensemble TL is more effective.

Figs. 9 and 10 display the loss and accuracy results of the proposed ensemble transfer learning model for detecting zero-day botnets when applied to the Car-Hacking and N-BaIoT datasets, respectively. The results suggest that the ensemble TL model exhibited proficient learning abilities from both datasets. The experiment was carried out over a span of 50 epochs. The proposed ensemble technique achieved a validation accuracy of 100 % and a loss of 2.0 % when used to Car-Hacking. However, while evaluating the N-BaIoT data, the model achieved a validation accuracy of 99.99 % and a loss of 6.0 %.

3.2.1. Comparison with the state-of-the-art in binary scenario on intra-vehicle network traffic (Car-Hacking data)

Furthermore, the performance of the proposed models for zero-day botnet detection in IoV is evaluated in comparison to other cutting-edge techniques [14,15,22,29]. Table 7 demonstrates that the majority of the analyzed methods obtain a high level of accuracy, which can be attributed to the straightforward nature of the Car-Hacking dataset. Out of the models presented in Table 7, the proposed ensemble TL approach demonstrate superior performance by reaching perfect scores of 100 % in accuracy, recall, precision, F1-score, and MCC. The majority of existing researchers did not take into account the use of MCC, training time, and testing time as a means of evaluating their performance.

3.3. Proposed ensemble TL performance analysis in binary scenario on external-vehicular network traffic

The optimized TL-based model for zero-day botnet detection on the N-BaIoT achieve exceptional performance, with accuracy, recall, precision, F1-scores, and MCC above 99.9 % following the implementation of BMO. However, it should be noted that Inception and ResNet models do not exhibit the same level of performance, as indicated in Table 8. The suggested ensemble TL model (soft voting) achieves a perfect accuracy of 100 %, a recall rate of 99.99 %, a precision rate of 99.99 %, an F1-score of 100 %, and a Matthews correlation coefficient of 100 %.

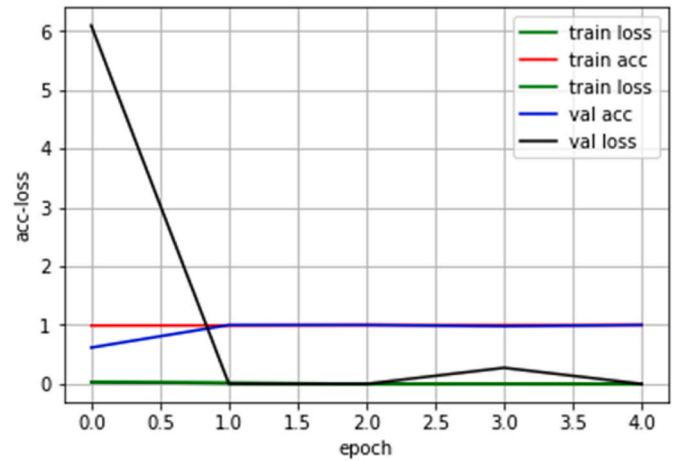


Fig. 10. The training accuracy, training loss versus validation accuracy and validation loss on the external vehicular network.

3.3.1. Comparison with the state-of-the-art in binary scenario on external-vehicular network traffic

The proposed ensemble TL model for zero-day botnet detection demonstrates superior performance compared to other recent methods mentioned in Table 9 [27,63]. Furthermore, the total duration of training and testing for the recent methods mentioned in the literature is very high. Additionally, these existing techniques failed to consider the Mathew Correlation Coefficient in evaluating the performance of their models.

3.4. Proposed ensemble TL performance analysis in multi-class detection scenario on intra-vehicle network traffic (Car-Hacking data)

Table 10 showcases the results of the Ensemble TL, which has been optimized using BMO, on the Car-Hacking data. The ensemble TL model successfully identified the benign sample in the multi-class situation, demonstrating efficient detection without any false positives. The

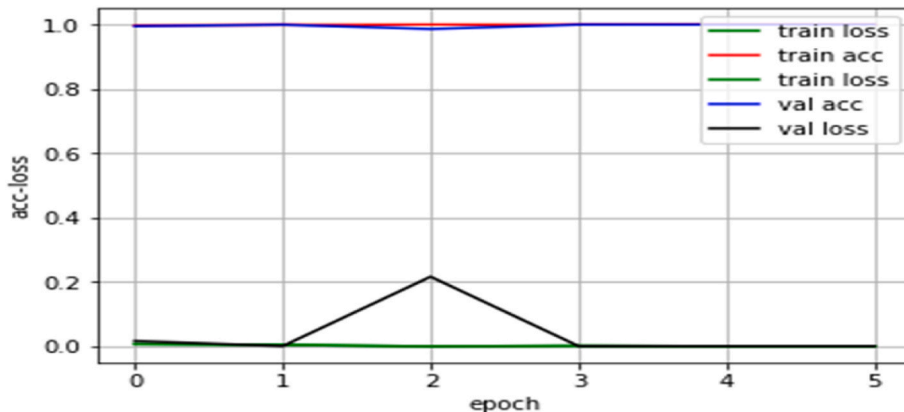


Fig. 9. The training accuracy, training loss versus validation accuracy and validation loss on the Car-Hacking data.

Table 7

Comparative analysis of the proposed Ensemble TL model with the state-of-the-art Intra-vehicle Network Traffic (Car-Hacking data).

Method	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
OneD-CNN [15]	99.96 %	99.63 %	99.94 %	99.80 %	×	×	×
LeNet [14]	98.10 %	98.04 %	98.14 %	97.83 %	×	×	×
DeepCNN [22]	99.93 %	99.84 %	99.84 %	99.91 %	×	×	×
Ensemble CNN [29]	100 %	100 %	100 %	100 %	×	2490.5	3.2
EnsembleTL (Soft voting)	100 %	100 %	100 %	100 %	100 %	1790.8	2.9

Table 8

Performance evaluation of optimized TL models and Ensemble TL models on N-BaIoT.

Methods	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
ResNet-BMO	99.80	99.85	99.78	99.65	99.55	160.8	1.2
Inception ResNet-BMO	99.98	99.99	99.89	99.89	99.75	615.2	2.1
Inception-BMO	99.70	99.82	99.70	99.51	99.65	183.1	1.9
MobileNet-	99.99	99.99	99.09	99.92	99.92	703.9	3.0
EnsembleTL (Soft voting)	100	99.99	99.99	100	100	2010.1	3.5

Table 9

Comparative analysis of the proposed Ensemble TL model with the state-of-the-art extra-vehicle network traffic.

Method	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
RF [63]	98.82 %	99.95	98.80 %	98.80 %	×	1848.3	0.3
KNN [63]	96.3 %	93.7 %	96.2 %	96.3 %	×	15243.6	0.2
MLP [27]	99.46 %	99.40 %	99.52 %	99.46 %	×	×	1.1
Ensemble CNN[29]	99.899 %	99.89 %	99.90 %	99.89 %	×	3598.7	1.8
Ensemble TL (Soft voting)	100 %	99.99 %	99.99 %	100 %	100 %	2010.1	3.5

Table 10

Performance evaluation of the proposed Ensemble TL models on Car-Hacking in multi-class scenario.

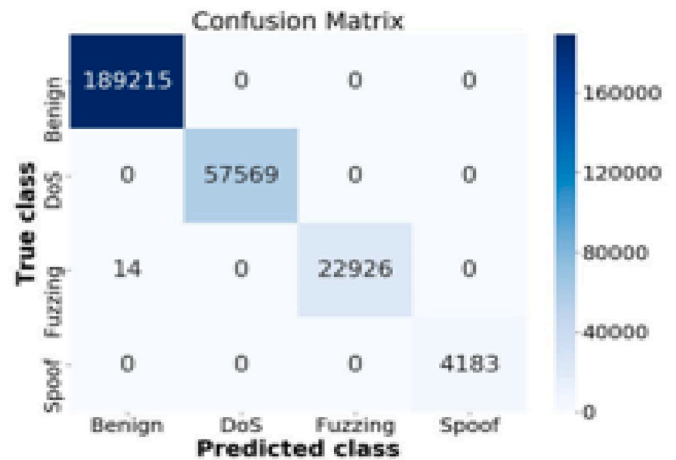
Label/Metric	Accuracy	Recall	Precision	F1-score	MCC	Training time	Testing time
Normal	100	100	100	100	100	1020	2.7
Gear Spoofing	99.99	99.92	100	99.94	99.95	120.1	1.9
Fuzzy	100	100	99.97	100	100	115.9	0.9
DoS	100	100	99.08	100	100	118.2	0.7
RPM Spoofing	99.99	99.99	99.99	99.97	99.98	103.2	2.1
Average	99.99	99.97	99.76	99.97	99.98	114.3	1.4

ensemble transfer learning model achieved a classification accuracy of above 99 % on the malicious sample with a multi-class situation. Nevertheless, the model inaccurately identified approximately 0.08 %–0.01 % of samples in the gear spoofing and RPM spoofing categories. The primary cause of this misclassification can be traced to the imbalanced distribution of classes in the training set. In summary, the suggested Ensemble TL model is highly effective in detecting the majority of zero-day attacks on intra-vehicle networks, with the exception of gear spoofing and RPM spoofing attacks.

The evaluation of the proposed ensemble transfer learning-based zero-day botnet attack detection is undertaken using the confusion matrix (CM) [64]. The CM matrix is a commonly used tool in the field of ML for assessing the effectiveness of ML algorithms. This offers a clear explanation of the classifier's capacity to accurately recognize the samples that pertain to separate categories within the dataset. Figs. 11 and 12 illustrate the confusion matrix of the ensemble-TL method for detecting zero-day botnet attacks on the Car-Hacking and N-BaIoT datasets, respectively.

3.4.1. Comparison with the state-of-the-art in multi-class scenario on intra-vehicle network traffic (Car-Hacking data)

The ensemble TL model developed for zero-day botnet detection has improved performance in comparison to other recently described methods outlined in Table 11 [20–22]. The authors in Ref. [20] model

**Fig. 11.** CM of the Car-Hacking data in multi-class scenario.

reached an accuracy of 97.4 % which still require more development in the accuracy. Additionally, the model employed two data sets, one for fuzzy attack and other one for DoS attack without factoring zero-day

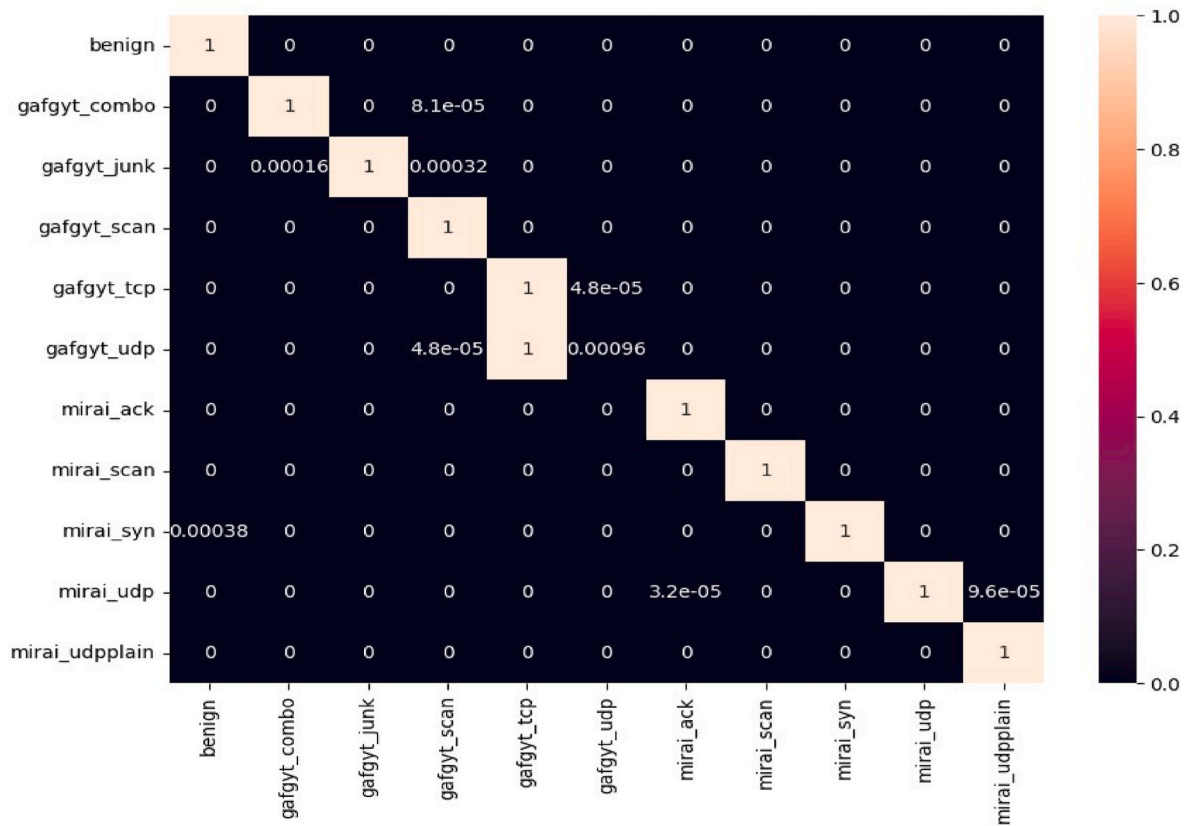


Fig. 12. CM of N-Balot in multi-class scenario.

Table 11

Comparative analysis of the proposed Ensemble TL model with the state-of-the-art In-vehicle network traffic in multi-class scenario.

Method	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
Abdulazeez et al. [20],	97.4 %	96.3 %	x	93.41 %	x	911.6	x
Habeeb et al. [21],	87.21 %	86.66 %	x	92.1 %	x	x	x
Song et al. [22],	99.93 %	99.84 %	x	99.91	x	x	x
Ashraf et al.[25]	99.0	99.0	x	99.0	x	x	x
Ensemble TL (Soft voting)	99.99 %	99.97 %	99.76	99.97 %	99.98 %	114.3	1.4

Table 12

Performance evaluation of the proposed Ensemble TL models on external-vehicular traffic in multi-class scenario.

Scenario	Classes	Accuracy	Recall	Precision	F1-score	MCC	Training time(sec)	Testing Time(sec)
Binary	Attack	100	100	100	100	100	130.2	5.23
	Benign	100	100	100	100	100	100.9	4.02
3-class	Average	100	100	100	100	100	115.55	4.62
	Mirai	99.98	100	99.90	99.96	99.97	234.3	4.52
	Gafgyt	99.99	100	99.95	99.99	100	198.2	3.41
	Benign	100	100	100	100	100	110.2	4.23
	Average	99.99	100	99.95	99.98	99.99	180.9	4.05
9-class	Benign	100	100	100	100	100	109.3	3.62
	ACK	99.97	99.97	99.96	99.99	99.89	121.2	3.42
	Junk	99.96	99.99	99.89	99.98	99.99	181.9	2.24
	Combo	99.99	99.99	99.70	99.97	99.99	203.2	3.25
	SYN	99.90	100	100	99.99	99.99	210.4	3.80
	Scan	100	100	99.98	99.99	100	219.3	2.98
	User Datagram Protocol	99.89	100	99.99	99.69	99.89	195.3	3.45
	TCP	99.56	99.99	99.98	99.80	99.99	186.2	2.23
	UPDPlain	99.80	99.90	99.96	99.99	99.09	172.4	3.34
	Average	99.88	99.98	99.93	99.92	99.85	186.24	3.08

botnet attack. The authors [21] obtained an accuracy of 87.21 %, a recall of 86.6 %, and a F1-score of 92.1 % without taking into account the MCC and precision. Furthermore, the study did not take into account the duration of training and testing. The proposed model [22] achieved an accuracy rate of 99.93 %, a recall rate of 99.84 %, and an F1-score of 99.91 %. However, the authors did not take into account the MCC and precision. Furthermore, the proposed model is a complex and opaque model, making it challenging to comprehend its interpretability. Additionally, the existing models in Table 11 did not consider zero-day botnet attack scenario.

3.5. Proposed ensemble TL performance analysis in multi-class detection scenario on external vehicular network traffic

Table 12 showcases the results of the XAI Ensemble TL, which has been optimized using BMO, on the external vehicle data. In the binary scenario, the ensemble TL model successfully identified the benign and malicious samples in the multi-class situation, demonstrating efficient detection without any false positives. In the scenario with 3 classes, the proposed ensemble TL model, achieved an accuracy rate of 99.99 %, recall of 100 %, F1-score of 99.98 %, and MCC of 99.99 %. Hence, the training time is about 180.9 s which is higher than the binary classification scenario. In the case of 9 classes, the proposed ensemble transfer learning model achieved an accuracy rate of 99.88 %, precision of 99.93 %, recall of 99.98 %, F1-score of 99.92 %, and MCC of 99.85 %.

The model successfully identifies various sorts of attacks; however, it has greater rates of failing to detect ACK, junk, Combo, TCP, and UDP plain attacks. The results demonstrate that the propose approach is generally efficient in detecting network intrusions. Nevertheless, their ability to classify varies based on the specific sort of attack. IoV systems may be more vulnerable to botnet assaults and intrusions due to the higher occurrence of false negative rates in certain attack categories. Therefore, it is crucial to do feature dimensionality reduction and include more instances of these botnet attack categories in the training set. Alternatively, one may explore using a data-level strategy, such as under sampling, to improve the balance of the training set.

3.5.1. Comparison with the state-of-the-art in multi-class scenario on external-vehicular network traffic

The XAI ensemble TL model, designed for detecting zero-day botnets, exhibits superior performance when compared to other previously proposed approaches as detailed in Table 13. The authors [63] have developed a dependable model and dataset that includes benign and seven prevalent attack network flows. This model and dataset adhere to real-world standards and are publicly accessible. The model attained an accuracy rate of 96.3 %, a recall rate of 96.2 %, and an F1-score of 96.3 %. The SU-IDS system was created by the authors [65] for the purpose of network intrusion detection using both semi-supervised and unsupervised techniques. The model attained a rate of accuracy of 99.13 %, and a recall rate of 99.65. The authors [66] have presented a new intrusion detection method for wireless mesh networks. This system utilizes GA-based feature selection and multiple SVM classifiers. The model attained an accuracy rate of 98.55 %, a recall rate of 98.22 %, and an F1-score of 98.3 %. Most datasets used in prior studies to train the models do not contain real-world network traffic relevant to IoV networks. Furthermore, most of the current solutions have only focused on

the network traffic within the vehicle, while disregarding the network traffic from other sources. Furthermore, the bulk of the current models are unsuitable for detecting zero-day botnets on the IoV. Most anomalous intrusion detection systems in the existing state-of-the-art models are not suited for deployment in a production setting, and cannot be implemented on an IoV gateway electronic control unit.

3.5.2. Novel Insights gained from comparison with the state-of-the-art models

3.5.2.1. Superior performance of the proposed model.

- Significance:** The results demonstrate that our proposed XAI ensemble transfer learning model outperforms state-of-the-art methods in detecting zero-day botnet attacks in IoV networks. Specifically, the model achieves higher accuracy, precision, recall, and MCC, which are critical for ensuring robust detection of sophisticated, unseen threats.
- Comparison:** In comparison to traditional deep learning models like standard CNNs in Ref. [22] and standalone TL architectures, our XAI ensemble model consistently delivers improved results across all key metrics. Existing methods often struggle with the detection of zero-day botnets due to limited generalization capabilities when applied to unseen attacks, whereas our model demonstrates better adaptability and robustness.
- Novel Insights:** The integration of eXplainability via XAI not only improves performance but also provides insights into the model's decision-making process. This is a novel addition compared to existing methods, which generally function as black boxes and do not offer transparency into why certain decisions are made.

3.6. Conclusion and recommendation

Efficient and optimized IDS have become crucial security applications in current systems like the Internet of Vehicles (IoV). Numerous adversaries target these vehicles, resulting in detrimental effects on their performance, posing risks to public and private property, and compromising road safety. Although there have been numerous studies on the design of IDS for vehicular networks, most of them focus solely on detecting known attacks within the vehicle itself or on external networks. In addition, a limited number of articles focus only on a certain sort of attack, such as DoS or spoofing attacks. Nevertheless, in practical scenarios, both the internal networks within vehicles and the external networks they connect to are susceptible to a wide range of attacks, including both known and emerging zero-day attack patterns. Therefore, it is necessary to develop an IDS that can identify both previously unknown attacks (zero-day attacks) and known attacks on both the internal vehicle network and the external vehicular networks. However, in order to ensure the reliability and practicality of IDSs in real-world vehicle systems, it is crucial to assess the comprehensibility and interpretability of the model. Unfortunately, previous research has not addressed this aspect, leaving the feasibility of other techniques in real-world IoVs unproven. Hence, in this research, we design a

Table 13

Comparative analysis of the proposed XAI Ensemble TL model with the state-of-the-art External-vehicle network traffic in multi-class scenario.

Method	Accuracy	Recall	Precision	F1-score	MCC	Training time (sec)	Testing time (sec)
Sharafaldin et al. [63]	96.3	96.2	x	96.3	x	15243.6	x
Min et al. [65]	99.13	99.65	x	x	x	x	x
Vijayanand et al. [66]	98.55	98.22	x	98.3	x	34896.5	x
Yang et al.[28]	99.86	99.75	x	99.86	x	5238.4	x
Ensemble TL (Soft voting)	99.88	99.98	99.93	99.92	99.85	186.24	3.08

comprehensive explainable ensemble transfer learning for zero-day botnet intrusion detection in IoV. We performed a high-dimensionality analysis of IoV network traffic using hybrid bidirectional long-short-term memory (BiLAE). In the second stage, we examined the Deep Shapley Explainable AI (XAI) to determine its explainability. Next, we implemented model hyper-parameter optimization on each of the ResNet, Inception, Inception ResNet, and mobileNet CNN-transfer learning architectures. Finally, we trained the ensemble CNN-TL on both intra-vehicular (CAN-hacking) and external vehicular (N-BaIoT) data. Findings from the simulation experiments showed that the XAI ensemble TL model for zero-day botnets, works better on an internal vehicular network. It got an accuracy rate of 100 %, recall of 100 %, precision of 100 %, F1-score of 100 %, and MCC of 100 % in a binary-class scenario. The XAI ensemble TL model exhibits superior performance when compared to other previously state-of-the-art approaches. The proposed XAI ensemble TL model is suited for deployment in a production setting, and can be implemented on an IoV gateway electronic control unit. The future work will be to integrate the proposed explainable ensemble transfer learning model IDS framework with blockchain to enhance user privacy in the IoV network.

CRedit authorship contribution statement

Yakub Kayode Saheed: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization, Software. **Joshua Ebere Chukwuere:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

- [1] X. Li, Z. Hu, M. Xu, Y. Wang, J. Ma, Transfer learning based intrusion detection scheme for Internet of vehicles, *Inf. Sci.* 547 (2021) 119–135, <https://doi.org/10.1016/j.ins.2020.05.130>.
- [2] A.Q. Adeyiola, Y.K. Saheed, S. Misra, S. Chockalingam, Metaheuristic firefly and C5.0 algorithms based intrusion detection for critical infrastructures, in: 2023 3rd International Conference on Applied Artificial Intelligence (ICAPAI), 2023, pp. 1–7, <https://doi.org/10.1109/ICAPAI58366.2023.10193917>.
- [3] H. Liang, et al., Network and system level security in connected vehicle applications, *IEEE/ACM Int. Conf. Comput. Des. Dig. Tech. Pap. ICCAD* (2018), <https://doi.org/10.1145/3240765.3243488>.
- [4] M. Gmiden, M.H. Gmiden, H. Trabelsi, An intrusion detection method for securing in-vehicle CAN bus, in: 2016 17th Int. Conf. Sci. Tech. Autom. Control Comput. Eng. STA 2016 - Proc., 2017, pp. 176–180, <https://doi.org/10.1109/STA.2016.7952095>.
- [5] J. Liu, S. Zhang, W. Sun, Y. Shi, In-vehicle network attacks and countermeasures: challenges and future directions, *IEEE Netw* 31 (5) (2017) 50–58, <https://doi.org/10.1109/MNET.2017.1600257>.
- [6] O.H. Abdulganiyu, T.A. Tchakoucht, Y.K. Saheed, Towards an efficient model for network intrusion detection system (IDS): systematic literature review, *Wirel. Netw* (2023), <https://doi.org/10.1007/s11276-023-03495-2>.
- [7] O.Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, A. Mouzakitis, Intrusion detection systems for intra-vehicle networks: a review, *IEEE Access* 7 (2019) 21266–21289, <https://doi.org/10.1109/ACCESS.2019.2894183>.
- [8] K. Koscher, et al., Experimental security analysis of a modern automobile, *Proc. - IEEE Symp. Secur. Priv.* (2010) 447–462, <https://doi.org/10.1109/SP.2010.34>.
- [9] Q. Wang, Y. Qian, Z. Lu, Y. Shoukry, G. Qu, A delay based plug-in-monitor for intrusion detection in controller area network, in: *Proc. 2018 Asian Hardw. Oriented Secur. Trust Symp. AsianHOST* 2018, 2019, pp. 86–91, <https://doi.org/10.1109/AsianHOST.2018.8607178>.
- [10] E. Seo, H.M. Song, H.K. Kim, GIDS: GAN based intrusion detection system for in-vehicle network, in: 2018 16th Annu. Conf. Privacy, Secur. Trust. PST, 2018, <https://doi.org/10.1109/PST.2018.8514157>, 0–5, 2018.
- [11] Y.K. Saheed, S. Misra, A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things, *Int. J. Inf. Secur.* 23 (3) (2024) 1557–1581, <https://doi.org/10.1007/s10207-023-00803-x>.
- [12] S. Abdelkader, et al., Securing modern power systems: implementing comprehensive strategies to enhance resilience and reliability against cyber-attacks, *Results Eng* 23 (May) (2024) 102647, <https://doi.org/10.1016/j.rineng.2024.102647>.
- [13] L. Yang, A. Moubayed, I. Hamieh, A. Shami, Tree-based intelligent intrusion detection system in internet of vehicles, in: 2019 IEEE Glob. Commun. Conf. GLOBECOM 2019 - Proc., No. ML, 2019, <https://doi.org/10.1109/GLOBECOM38437.2019.9013892>.
- [14] K. Mehedi, S.T. A. Anwar, Z. Rahman, Ahmed, Deep transfer learning based intrusion detection system for electric vehicular networks, *Sensors* 21 (14) (2021) 1–23, <https://doi.org/10.3390/s21144736>.
- [15] D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, An Effective In-Vehicle CAN Bus Intrusion Detection System Using CNN Deep Learning Approach (2020), *GLOBECOM 2020 - 2020 IEEE Global Communications Conference, Taipei, Taiwan*, 2020, pp. 1–6, <https://doi.org/10.1109/GLOBECOM42002.2020.9322395>.
- [16] B. Zadrozny, Learning and evaluating classifiers under sample selection bias, in: *Proceedings, Twenty-First Int. Conf. Mach. Learn. ICML, 2004*, pp. 903–910, <https://doi.org/10.1145/1015330.1015425>, 2004.
- [17] Y. Kayode Saheed, A. Idris Abiodun, S. Misra, M. Kristiansen Holone, R. Colomo-Palacios, A machine learning-based intrusion detection for detecting internet of things network attacks, *Alexandria Eng. J.* 61 (12) (2022) 9395–9409, <https://doi.org/10.1016/j.aej.2022.02.063>.
- [18] O.H. Abdulganiyu, T. Ait Tchakoucht, Y.K. Saheed, A systematic literature review for network intrusion detection system (IDS), *Int. J. Inf. Secur.* (2023), <https://doi.org/10.1007/s10207-023-00682-2>.
- [19] M.J. Kang, J.W. Kang, Intrusion detection system using deep neural network for in-vehicle network security, *PLoS One* 11 (6) (2016) 1–17, <https://doi.org/10.1371/journal.pone.0155781>.
- [20] A. Alshammari, M.A. Zohdy, D. Debnath, G. Corser, Classification approach for intrusion detection in vehicle systems, *Wirel. Eng. Technol.* 9 (4) (2018) 79–94, <https://doi.org/10.4236/wet.2018.94007>.
- [21] H. Olufowobi, C. Young, J. Zambreno, G. Bloom, SAIDuCANT: specification-based automotive intrusion detection using controller area network (CAN) timing, *IEEE Trans. Veh. Technol.* 69 (2) (2020) 1484–1494, <https://doi.org/10.1109/TVT.2019.2961344>.
- [22] H.M. Song, J. Woo, H.K. Kim, In-vehicle network intrusion detection using deep convolutional neural network, *Veh. Commun.* 21 (2020) 100198, <https://doi.org/10.1016/j.vehcom.2019.100198>.
- [23] Y. Yang, Z. Duan, M. Tehranipoor, Identify a spoofing attack on an in-vehicle can bus based on the deep features of an ecu fingerprint signal, *Smart Cities* 3 (1) (2020) 17–30, <https://doi.org/10.3390/smartcities301002>.
- [24] M.D. Hossain, H. Inoue, H. Ochiai, D. Fall, Y. Kadobayashi, LSTM-based intrusion detection system for in-vehicle can bus communications, *IEEE Access* 8 (2020) 185489–185502, <https://doi.org/10.1109/ACCESS.2020.3029307>.
- [25] J. Ashraf, A.D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, A. Beheshti, Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems, *IEEE Trans. Intell. Transp. Syst.* 22 (7) (2021) 4507–4518, <https://doi.org/10.1109/TITS.2020.3017882>.
- [26] K.M.A. Alheeti, K. McDonald-maier, Intelligent intrusion detection in external communication systems for autonomous vehicles autonomous vehicles, *Syst. Sci. Control Eng.* 6 (1) (2018) 48–56, <https://doi.org/10.1080/21642583.2018.1440260>.
- [27] A. Rosay, P. Leroux, *Feed-forward Neural Network for Network Intrusion Detection*, vol. 48, 2020.
- [28] L. Yang, A. Moubayed, A. Shami, MTH-IDS: a multitiered hybrid intrusion detection system for internet of vehicles, *IEEE Internet Things J.* 9 (1) (2022) 616–632, <https://doi.org/10.1109/JIOT.2021.3084796>.
- [29] L. Yang, A. Shami, A transfer learning and optimized CNN based intrusion detection system for internet of vehicles, *IEEE Int. Conf. Commun.* (2022) 2774–2779, <https://doi.org/10.1109/ICC45855.2022.9838780>, 2022-May.
- [30] Y. Kayode Saheed, O. Harazeem Abdulganiyu, T. Ait Tchakoucht, A novel hybrid ensemble learning for anomaly detection in industrial sensor networks and SCADA systems for smart city infrastructures, *J. King Saud Univ. - Comput. Inf. Sci.* 35 (5) (2023) 101532, <https://doi.org/10.1016/j.jksuci.2023.03.010>.
- [31] K.M. Ali Alheeti, K. Mc Donald-Maier, Intelligent intrusion detection in external communication systems for autonomous vehicles, *Syst. Sci. Control Eng.* 6 (1) (2018) 48–56, <https://doi.org/10.1080/21642583.2018.1440260>.
- [32] M. Aloqaily, S. Otoum, I. Al Ridhawi, Y. Jararweh, An intrusion detection system for connected vehicles in smart cities, *Ad Hoc Netw.* 90 (2019), <https://doi.org/10.1016/j.adhoc.2019.02.001>.
- [33] S. Jafari, J.H. Yang, Y.C. Byun, Optimized XGBoost modeling for accurate battery capacity degradation prediction, *Results Eng* 24 (August) (2024) 102786, <https://doi.org/10.1016/j.rineng.2024.102786>.
- [34] M. Turki Almousa, M.R. Gomaa, M. Ghasemi, M. Louzazni, Gradient-based optimization for parameter identification of lithium-ion battery model for electric vehicles, *Results Eng* 24 (August) (2024) 102845, <https://doi.org/10.1016/j.rineng.2024.102845>.
- [35] A.H. Sodhro, G.H. Sodhro, M. Guizani, S. Pirbhulal, A. Boukerche, AI-enabled reliable channel modeling architecture for fog computing vehicular networks, *IEEE*

- Wirel. Commun. 27 (2) (2020) 14–21, <https://doi.org/10.1109/MWC.001.1900311>.
- [36] R. Suganya, L.M.I.L. Joseph, S. Kollem, Understanding lithium-ion battery management systems in electric vehicles : environmental and health impacts , comparative study , and future trends : a review, Results Eng 24 (September) (2024) 103047, <https://doi.org/10.1016/j.rineng.2024.103047>.
- [37] P. Shrivastava, M.S. Jamal, K. Kataoka, EvilScout: detection and mitigation of evil twin attack in SDN enabled WiFi, IEEE Trans. Netw. Serv. Manag. 17 (1) (2020) 89–102, <https://doi.org/10.1109/TNSM.2020.2972774>.
- [38] Y.K. Saheed, A.I. Omole, M.O. Sabit, GA-mADAM-IIoT: a new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data, Sensors Int 6 (August 2024) (2025) 100297, <https://doi.org/10.1016/j.sintl.2024.100297>.
- [39] S.I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, O. Jogunola, Federated deep learning for zero-day botnet attack detection in IoT-edge devices, IEEE Internet Things J. 9 (5) (2022) 3930–3944, <https://doi.org/10.1109/JIOT.2021.3100755>.
- [40] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: bot-IoT dataset, Futur. Gener. Comput. Syst. 100 (2019) 779–796, <https://doi.org/10.1016/j.future.2019.05.041>.
- [41] N. Kaja, Artificial Intelligence and Cybersecurity: Building an Automotive Cybersecurity Framework Using Machine Learning Algorithms, Univ. Michigan-Dearborn, 2019, pp. 1–140.
- [42] D.A. Schmidt, M.S. Khan, B.T. Bennett, Spline-based intrusion detection for VANET utilizing knot flow classification, Internet Technol. Lett. 3 (3) (2020) 2–7, <https://doi.org/10.1002/itl2.155>.
- [43] P. Rosay, A. Carlier, F. Leroux, MLP4NIDS: An Efficient MLP-Based Network Intrusion Detection for CICIDS2017 Dataset (2020).
- [44] Y.K. Saheed, M.A. Raji, Effectiveness of deep learning long short-term memory network for stock price prediction on graphics processing unit, in: 2022 International Conference on Decision Aid Sciences and Applications (DASA), 2022, pp. 1665–1671, <https://doi.org/10.1109/DASA54658.2022.9765181>.
- [45] F. Zhao, J. Feng, J. Zhao, W. Yang, S. Yan, Robust LSTM-autoencoders for face de-occlusion in the wild, IEEE Trans. Image Process. 27 (2) (2018) 778–790, <https://doi.org/10.1109/TIP.2017.2771408>.
- [46] S. Hochreiter, Long Short-Term Memory, vol. 1780, 1997, pp. 1735–1780.
- [47] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proc. IEEE 78 (10) (1990) 1550–1560, <https://doi.org/10.1109/5.58337>.
- [48] J.F. Torres, F. Martínez-Álvarez, A. Troncoso, A deep LSTM network for the Spanish electricity consumption forecasting, Neural Comput. Appl. 34 (13) (2022) 10533–10545, <https://doi.org/10.1007/s00521-021-06773-2>.
- [49] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural Network. 18 (5–6) (2005) 602–610, <https://doi.org/10.1016/j.neunet.2005.06.042>.
- [50] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681, [https://doi.org/10.1016/s1634-6939\(13\)59289-1](https://doi.org/10.1016/s1634-6939(13)59289-1).
- [51] M.M. Leonardo, T.J. Carvalho, E. Rezende, R. Zucchi, F.A. Faria, Deep feature-based classifiers for fruit fly identification (Diptera: tephritidae), in: Proc. - 31st Conf. Graph. Patterns Images, SIBGRAPI 2018, 2018, pp. 41–47, <https://doi.org/10.1109/SIBGRAPI.2018.00012>.
- [52] D. Petrov, T.M. Hospedales, Measuring the transferability of adversarial examples [Online]. Available: <http://arxiv.org/abs/1907.06291>, 2019.
- [53] K. Zhang, Y. Zhang, M. Wang, A unified approach to interpreting model predictions scott, NIPS (News Physiol. Sci.) 16 (3) (2012) 426–430.
- [54] J. Zhang, F. Li, H. Zhang, R. Li, Y. Li, Intrusion detection system using deep learning for in-vehicle security, Ad Hoc Netw. 95 (2019) 101974, <https://doi.org/10.1016/j.adhoc.2019.101974>.
- [55] C. Molnar, Interpretable machine learning. A guide for making black box models explainable, Book (2020) 247 [Online]. Available: <https://christophm.github.io/interpretable-ml-book>.
- [56] Y.K. Saheed, T.T. Salau-Ibrahim, M. Abdulsalam, I.A. Adeniji, B.F. Balogun, Modified bi-directional long short-term memory and hyperparameter tuning of supervised machine learning models for cardiovascular heart disease prediction in mobile cloud environment, Biomed. Signal Process Control 94 (February) (2024) 106319, <https://doi.org/10.1016/j.bspc.2024.106319>.
- [57] M.H. Sulaiman, Z. Mustafa, M.M. Saari, H. Daniyal, Barnacles Mating Optimizer: a new bio-inspired algorithm for solving engineering optimization problems, Eng. Appl. Artif. Intell. 87 (October 2019) (2020) 103330, <https://doi.org/10.1016/j.engappai.2019.103330>.
- [58] L. Khandeparker, A.C. Anil, Underwater adhesion: the barnacle way, Int. J. Adhes. Adhes. 27 (2) (2007) 165–172, <https://doi.org/10.1016/j.ijadhadh.2006.03.004>.
- [59] Y.K. Saheed, O.H. Abdulganiyu, K.U. Majikumna, M. Mustapha, A.D. Workneh, ResNet50-1D-CNN: a new lightweight resNet50-One-dimensional convolution neural network transfer learning-based approach for improved intrusion detection in cyber-physical systems, Int. J. Crit. Infrastruct. Prot. 45 (March) (2024) 100674, <https://doi.org/10.1016/j.ijcip.2024.100674>.
- [60] Y.K. Saheed, O.H. Abdulganiyu, T.A. Tchakoucht, Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities, Appl. Soft Comput. 155 (February) (2024) 111434, <https://doi.org/10.1016/j.asoc.2024.111434>.
- [61] M.Z. Naser, An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: navigating causality, forced goodness, and the false perception of inference, Autom. Constr. 129 (July) (2021) 103821, <https://doi.org/10.1016/j.autcon.2021.103821>.
- [62] G. yao Zhao, et al., Enhancing interpretability of tree-based models for downstream salinity prediction: decomposing feature importance using the Shapley additive explanation approach, Results Eng 23 (May) (2024) 102373, <https://doi.org/10.1016/j.rineng.2024.102373>.
- [63] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, Cic (2018) 108–116, <https://doi.org/10.5220/0006639801080116>.
- [64] T.M. Usman, Y.K. Saheed, A. Nsang, A. Ajibesin, S. Rakshit, A systematic literature review of machine learning based risk prediction models for diabetic retinopathy progression, Artif. Intell. Med. 143 (2023), <https://doi.org/10.1016/j.artmed.2023.102617>.
- [65] J. Min, E. J. Long, Q. Liu, J. Cui, Z. Cai, Ma, SU-IDS: a semi-supervised and unsupervised framework for, Network Intrusion Detection (2018), https://doi.org/10.1007/978-3-030-00012-7_30.
- [66] R. Vijayanand, D. Devaraj, B. Kannapiran, Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection, Comput. Secur. 77 (2018) 304–314, <https://doi.org/10.1016/j.cose.2018.04.010>.