Received 10 June 2024, accepted 30 July 2024, date of publication 2 August 2024, date of current version 12 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3437416

RESEARCH ARTICLE

Intrusion Detection System for Vehicular Networks Based on MobileNetV3

SHAOQIANG WANG, YIZHE WANG, BAOSEN ZHENG, JIAHUI CHENG, YU SU, AND YINFEI DAI
School of Computer Science and Technology, Changchun University, Changchun 130022, China

Corresponding author: Yinfei Dai (220701266@mails.ccu.edu.cn)

This work was supported by the Project of Jilin Provincial Scientific and Technological Department under Grant 20240101343JC and Grant 20220201154GX.

ABSTRACT With the advancement and refinement of intelligence and connectivity, intelligent connected vehicles have emerged as a prominent trend in contemporary development. Consequently, invasion attacks targeting these intelligent connected vehicles have also arisen. Mainstream intrusion detection systems (IDS) based on deep learning technologies can address malicious traffic infiltrations; however, they often fail to meet the real-time and lightweight requirements of vehicles. This paper introduces a lightweight vehicular intrusion detection method leveraging the MobileNetV3 architecture. By utilizing MobileNetV3 as the core framework, this method incorporates advanced techniques and design principles such as Depthwise Separable Convolution, Bottleneck structures, and Squeeze-and-Excitation (SE) modules. These innovations significantly reduce computational and parameter overhead while maintaining high model accuracy. Furthermore, MobileNetV3 is specifically designed for deployment on mobile devices, ensuring efficient operation even in resource-constrained environments. The proposed intrusion detection model achieved an accuracy, recall, precision, and F1 score of 100% on the Car-Hacking dataset, and an accuracy, recall, precision, and F1 score of 99.98% on the CICIDS-2017 dataset. The model size is 16MB. Experimental results demonstrate that this intrusion detection scheme not only accurately detects malicious attacks on vehicles but also meets the lightweight requirements of vehicular applications.

INDEX TERMS Intrusion detection, MobileNetV3, lightweight and efficient, connected vehicle networks.

I. INTRODUCTION

Since the Industrial Revolution in Britain, automobiles have come into public view, with the world's first car invented by the German Karl Benz. With the relentless march of technology, automotive electronic systems have seen exponential growth. Today's vehicular landscape has been dramatically reshaped; vehicles have evolved from purely mechanical, independent entities reliant solely on human intervention to entities that are markedly intelligent and electronic [1]. Consequently, vehicle communication systems based on the Internet of Things(IoT), also known as the Internet of Vehicles (IoV), have emerged [2]. In response, In this new era, vehicles are increasingly adopting electronic or electrical systems for control, with the Electronic Control Unit (ECU) playing a pivotal role in internal detection and control, thereby becoming the central authority managing various vehicular functions. These ECUs intercommunicate through the Controller Area Network (CAN), enabling a myriad of vehicular functionalities. Nonetheless, this technological progress brings with it significant security challenges, notably vulnerabilities within the CAN protocol that expose it to cyber attacks. Cyber attacks are primarily categorized into Intra-Vehicle Network (IVN) attacks and External Vehicle Network (EVN) attacks. IVN mainly target the ECUs, with the prevalent threats being message injection attacks caused by the lack of identity authentication in CAN communications and its unique broadcast transmission mechanism [3]. EVN, on the other hand, exploit weaknesses in the interactions between vehicles and infrastructures like roadside units

The associate editor coordinating the review of this manuscript and approving it for publication was Binit Lukose.

through the use of Vehicle-to-Everything (V2X) technology within Connected Vehicle Networks. Key attack vectors include Denial of Service attacks(DoS), sniffing, and spoofing attacks.

For Vehicle-to-Everything (V2X) in connected vehicles, the terminology encompasses Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Sensor (V2S), Vehicle-to-Roadside Unit (V2R), and Vehicle-to-Pedestrian (V2P) communications [4]. Figure 1 illustrates a practical transmission paradigm of V2X, which can further integrate additional technologies such as cloud and edge computing [5].

FIGURE 1. Vehicle-to-Everything (V2X) real-time transmission.

With the continuous maturation and advancement of vehicular network technology, the expansion of vehicle functionalities leads to an increase in information sharing among vehicles, which in turn elevates security risks, thereby diminishing the stability and robustness of the vehicles [6]. Within the broader context of cybersecurity, Intrusion Detection Systems (IDS) play a pivotal role. Traditional IDS methods include network-based, host-based, and hybrid architectures, utilizing signature detection, and anomaly detection techniques. However, these methods face limitations in meeting the requirements for real-time responsiveness and lightweight convenience demanded by vehicles. Consequently, an increasing number of researchers are leaning towards lightweight model approaches, offering new opportunities for intelligent intrusion detection systems.

Figure 2 displays a typical architecture designed to address attacks on vehicles. For Intra-Vehicle Network attacks, malicious CAN data traffic is launched against real vehicles through the Second-generation on-board diagnostics system (OBD II) port. The vehicle intrusion detection system is positioned between the external connections and the CAN bus, thereby filtering all data traffic entering the CAN bus. When an attacker attempts to inject malicious traffic into a vehicle, the Vehicle Intrusion Detection System will problem an alert message [7]. For External Vehicle Network attacks, they are primarily launched through a variety of wireless devices such as WiFi and Bluetooth. The vehicle intrusion detection system can be implemented as part of the gateway to identify and prevent these external malicious attacks targeting the vehicle [33].

Although significant strides have been made in the application of deep learning to IDS, numerous challenges persist. For example, when utilized in cybersecurity, deep learning models such as Recurrent Neural Networks (RNN) are required to process complex sequential data, which may lead to problems of vanishing or exploding gradients, thereby impacting the training and stability of the models. Traditional CNN models, such as VGGNet and AlexNet, typically possess a vast number of parameters, often exceeding hundreds of millions. This results in computationally intensive operations, particularly pronounced when dealing with large-scale or high-dimensional data. They require substantial computational resources, which are particularly inadequate in resource-constrained vehicular environments [8], [9]. Furthermore, the increasing complexity of modern cyber attacks has rendered traditional signature-based IDS methods increasingly ineffective, as they struggle to identify new or variant attacks [10], [11]. Particularly in Internet of Things (IoT) environments, the surge in attacks originating from numerous connected devices has made the challenges faced by traditional IDS methods even more pronounced. The resource constraints of IoT devices render the deployment of complex deep learning models on these devices impractical [12]. Moreover, although machine learning methods excel in certain scenarios, their capability is limited when it comes to handling zero-day attacks and the polymorphic nature of malware [11].

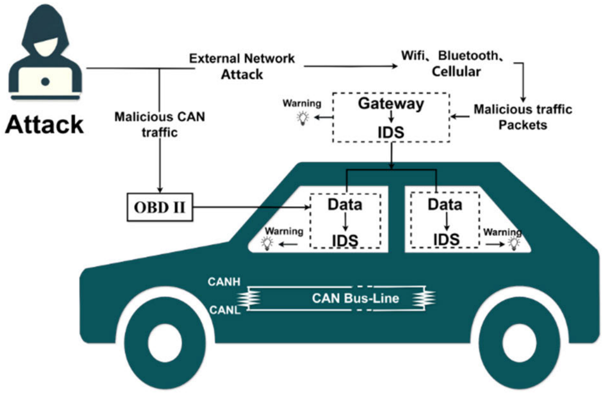


FIGURE 2. Architectural framework for internal and external vehicle attacks.

In this milieu, MobileNetV3, distinguished by its streamlined architecture, exhibits pronounced benefits in terms of efficiency, storage capacity, and power utilization. Utilizing depthwise separable convolutions, MobileNetV3 markedly diminishes the quantity of model parameters and computational burdens [13], thus reducing computational expenses and storage needs, whilst simultaneously augmenting processing velocity on edge devices [14].

To address the security vulnerabilities encountered during vehicle operation, this paper proposes a lightweight vehicle intrusion detection model based on MobileNetV3. This model is crafted to identify assaults on both the internal and external networks of vehicles, thus bolstering the precision of intrusion detection. This strategy signifies a groundbreaking shift from conventional machine-learning-based intrusion detection systems, facilitating more potent responses to the security challenges contemporary vehicles encounter. Initially, the proposed intrusion detection system processes an array of datasets to establish an optimal dataset. Thereafter, the cultivated dataset is partitioned into a training set and a testing set in an 80%-20% split. Ultimately, the segmented dataset is input into the intrusion detection model for both training and assessment. The primary contributions of this paper are outlined as follows:

1) We propose an anomaly detection method based on the Isolation Forest algorithm and a feature selection technique utilizing the Random Forest model. These methods are employed to eliminate irrelevant outliers from the dataset and to identify features pertinent to vehicle intrusion detection, thereby enhancing predictive performance.
2) A novel spatial feature extraction architecture has been designed to enhance the detection efficiency of the model by extracting spatial features from the dataset prior to training with the MobileNetV3 model.
3) Experimental evaluations were conducted on the partitioned dataset, and compared with other methodologies, our approach achieved superior results in terms of accuracy and F1 score.

The residual framework of this manuscript unfolds as follows. Initially, within Section II, we present the contemporary investigations pertaining to intrusion detection in vehicular networks. Subsequently, Section III scrutinizes the architectural design of the intrusion detection models elucidated in this paper. This encompasses preprocessing of data and the intrinsic architecture of the intrusion detection model. In Section IV, we utilize corresponding assessment methodologies to holistically evaluate the training outcomes of the model. Finally, in Section V, we concisely summarize the conclusions derived from this study and deliberate upon future work plans.

II. RELATED WORKS

Vehicular networks constitute an integrated network system that facilitates the exchange of information and collaborative control among vehicles, roadways, and other traffic participants through various sensors, controllers, and communication technologies [15]. Given its attributes such as elevated mobility and dynamism, vehicular ad hoc networks (VANETs) are prone to an extensive array of assaults [16]. The principal concerns encountered by VANETs encompass security, reliability, and confidentiality. To tackle these tripartite challenges, numerous scholars have conducted thorough explorations into VANETs. In their 2014 study, Engoulou et al. [16] put forward multiple strategies to mitigate the challenges inherent in vehicular ad hoc networks, However, there was an absence of discourse regarding methods for safeguarding privacy. Regarding the Controller Area Network (CAN), its deficiency lies in the absence of identity authentication and message source validation. Greenough demonstrated in their investigation [17] that vehicles can be remotely manipulated through attack vectors such as CD players and cellular networks, thereby exerting control over functions such as braking and steering whereas the vehicle is in motion, prompting widespread concern. Azees et al. scrutinized meticulously the methods of identity authentication for safeguarding privacy in their investigation of 2016 [18]. Due to their stringent requirements for real-time performance and reliability, as well as constraints in computational capacity, storage resources, and cost, In-Vehicle Networks (IVNs) pose considerable challenges [19]. Thus, in practical settings, conventional deep learning methodologies are frequently inapplicable to real-world IVN scenarios. In the research conducted by Wu et al. [19], a intrusion detection system based on temporal periodicity intervals was proposed. Nonetheless, this model is solely applicable for anomaly detection in cyclic message transmissions and lacks broad applicability. In the research conducted by Yao et al. [20], they introduced an intrusion detection model named STDeepGraph specifically designed for the extrinsic vehicular network. This model primarily focuses on malevolent data originating externally to the vehicle, The datasets employed encompass CICIDS2017 and UNSW-NB15, and the amalgamation of Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) architectures was utilized for this integration. In the research conducted by Aswal et al. [21], a proposition was made to employ classical machine learning algorithms for intrusion detection, utilizing the CICIDS2017 dataset. However, their consideration of various types of network attacks was deemed inadequate. In the research conducted by Schmidt et al. [22], a intrusion detection model named KFC was proposed, employing spline-based modeling. The NSL-KDD dataset was utilized for this endeavor. In the domain of intrusion detection, numerous investigations have harnessed a variety of machine learning and deep learning methodologies. For instance, certain studies concentrate on employing Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or their derivatives to discern anomalous patterns within network traffic. Whereas these approaches have demonstrated significant achievements in conventional network environments, they still face challenges in specific scenarios within the vehicular network domain, including problems related to processing speed, model complexity, and optimization of energy consumption. In the research conducted by Vuong et al. [23], they devised a decision tree-based intrusion detection model and conducted training to detect four types of attacks. The outcomes demonstrated notable precision, albeit with a latency of one second. In the research conducted by Fu et al. [24], they devised an intrusion detection system founded on FPGA architecture, attaining elevated real-time efficacy in vehicular settings. Nonetheless, this framework is strictly tailored for FPGA deployment and is ill-suited for alternative platform contexts, thereby harboring inherent constraints. In the investigation by Taylor et al. [25], they proposed a frequency-based detector capable of swiftly identifying anomalies by calculating the integral of curves within a one-second timeframe. However, this solution is confined to the usage of periodic messages, presenting considerable constraints within the context of vehicular networking environments. In the study conducted by Wasicek et al. [26], they proposed a behavior-centric context-aware intrusion detection framework, wherein the intrusion detection model framework utilizes networking mechanisms to discern operations. The study demonstrates that the model achieves an exceedingly high level of precision. However, its applicability to real-world scenarios or anticipated driving conditions may not be assured. In the domain of vehicular environments, the significance of lightweight models is paramount, owing to their diminished reliance on computational resources and expeditious processing velocity, thereby mitigating the constraints posed by the finite resources inherent in vehicular systems. In the realm of vehicular networking, particularly within the context of edge computing environments, employing lightweight models proximal to the data source facilitates expeditious processing, thereby mitigating both data transmission overhead and processing latency [27]. Additionally, the refinement and enhancement of streamlined models persistently captivate the interest of scholars. Specifically, through precise structural modifications and enhancements to the MobileNetV3 architecture, it is possible to substantially reduce the computational complexity while preserving high precision, an essential quality in the dynamically evolving environment of vehicular networks [13]. Moreover, by incorporating these optimized models into the security frameworks of vehicular networks, there is not only an augmentation of the network's real-time surveillance capabilities but also an improvement in the adaptability and reaction time to nascent threats [28]. By leveraging enhanced deep learning architectures, coupled with unique data from vehicular networks, effective pattern recognition and anomaly detection can be conducted within intricate network environments, thus significantly boosting the overall security efficacy of the network.

III. PROPOSED SYSTEM MODEL

A. FRAMEWORK OVERVIEW

In the current domain of vehicular malicious traffic data detection, numerous research efforts have aimed at developing effective intrusion detection systems. However, several shortcomings persist, presenting opportunities for our model framework. Firstly, many extant studies exhibit deficiencies in data preprocessing. Some research may be confined to rudimentary data cleaning and missing value imputation, overlooking more sophisticated preprocessing techniques. Secondly, certain existing models employ deep neural networks and ensemble learning methods characterized by excessive computational overhead and substantial memory requirements, limiting their applicability to theoretical research and impeding practical deployment in real-world vehicular systems.

Comparative Analysis with Existing Work, Many existing models suffer from limitations in both preprocessing and model complexity. While sophisticated methods like deep neural networks and ensemble learning are often employed, these approaches can be hindered by high computational costs and significant memory usage. For example, most of the experiments in the work used Convolutional Neural Network (CNN) based models. The main focus was on basic data cleaning techniques such as simply removing null values and linear interpolation of missing data. These models often neglect advanced preprocessing steps like SMOTE (Synthetic Minority Over-sampling Technique) for handling class imbalances or outlier detection using robust statistical methods. This can lead to suboptimal data quality and integrity, impacting the model's overall performance.

Addressing these deficiencies, our model framework offers distinct advantages. This study introduces a model framework specifically designed for detecting malicious traffic data in vehicular environments. The architecture of this vehicular intrusion detection system is depicted in Figure 3. Initially, we implement an extensive series of data preprocessing steps, including data filtering, outlier management, and oversampling, to ensure data quality and structural integrity. Unlike other studies that may perform basic preprocessing, our approach employs more targeted and innovative strategies in feature engineering, extracting richer and more representative features. For example, instead of merely imputing missing values, we use advanced techniques such as KNN imputation and robust outlier detection using the Isolation Forest method. This is followed by normalization and conversion into image format, ensuring that the dataset is optimal for model input and does not compromise detection efficacy. Subsequent to data preprocessing, we employ the MobileNetV3 model for training and evaluating the dataset. MobileNetV3 preserves model accuracy while significantly reducing computational and parameter overhead, making it suitable for practical deployment in resource-constrained vehicular environments. In contrast, most of the models proposed in the experiments using ensemble learning techniques such as random forests and XGBoost require a lot of computational resources and memory, making them less suitable for real-time vehicle applications. MobileNetV3 lightweight architecture, incorporating Depthwise Separable Convolutions and Squeeze-and-Excitation (SE) modules, allows it to perform efficiently with minimal resource consumption.

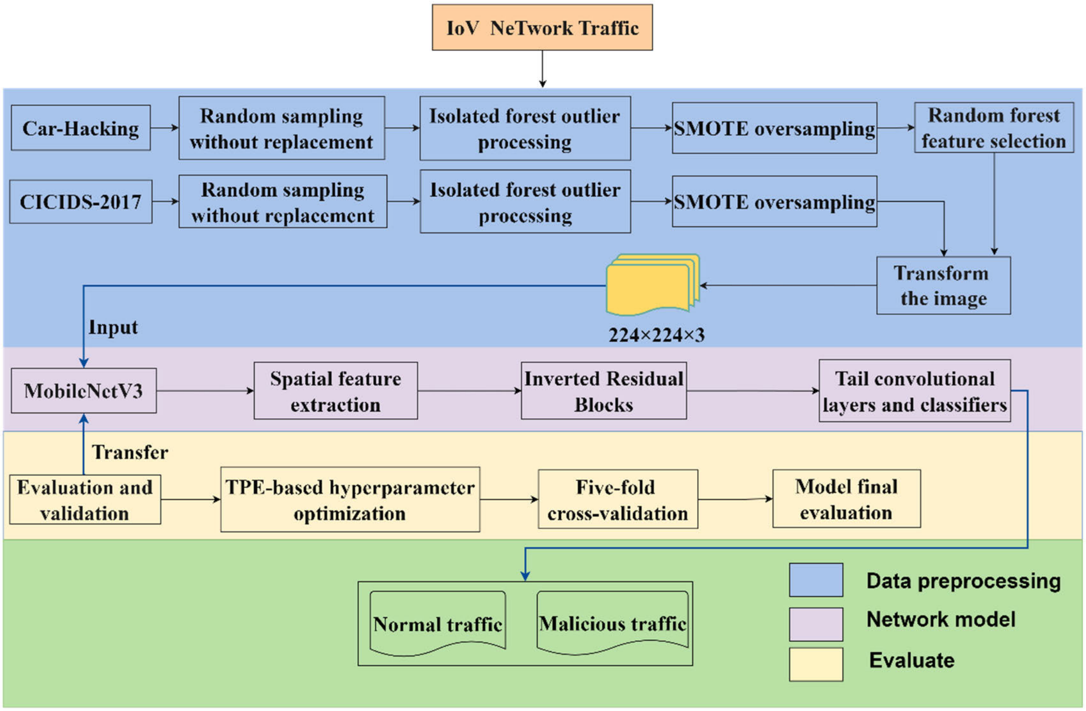

IoV NeTwork Traffic
Car-Hacking
Random sampling without replacement
Isolated forest outlier processing
SMOTE oversampling
Random forest feature selection
CICIDS-2017
Random sampling without replacement
Isolated forest outlier processing
SMOTE oversampling
Transform the image
224×224×3
Input
MobileNetV3
Spatial feature extraction
Inverted Residual Blocks
Tail convolutional layers and classifiers
Transfer
Evaluation and validation
TPE-based hyperparameter optimization
Five-fold cross-validation
Model final evaluation
Normal traffic
Malicious traffic
Data preprocessing
Network model
Evaluate

FIGURE 3. Proposed architectural model for vehicle intrusion detection systems.

To ensure the rigor of experimental results and the efficiency of the experimental process, the model undergoes spatial feature extraction, hyperparameter optimization, and five-fold cross-validation prior to final evaluation. These steps guarantee the model's accuracy and practicality, culminating in optimal experimental outcomes. For example, unlike the traditional models that may skip cross-validation due to computational constraints, our approach ensures robust model validation, enhancing reliability and performance metrics.

Our model framework surpasses traditional approaches in several aspects, particularly in preprocessing sophistication and computational efficiency. However, it is essential to acknowledge that while MobileNetV3 significantly reduces overhead, it may still face limitations in environments with extremely constrained resources. Future work could focus on further optimizing the model for such scenarios.

B. DATA PREPROCESSING

During efficient data analysis and model training, the data preprocessing stage plays a crucial role. This is especially true when employing advanced models like MobileNetV3, making this step particularly pivotal. The MobileNetV3 model, as a leading convolutional neural network (CNN), has demonstrated remarkable effectiveness in the field of image processing. The excellence of CNNs in image processing primarily lies in their ability to effectively identify and process spatial hierarchical structures within image data. Considering this, we have decided to convert the dataset into image format to better leverage the capabilities of the MobileNetV3 model. The dataset concerning both internal and external vehicle intrusions essentially comprises structured tabular data, rendering the conversion process into image data viable and expedient. This transformation involves several pivotal steps.

Initially, we need to load the entire network dataset. Subsequently, filtration and cleansing are conducted to ensure the quality and consistency of the data. Subsequently, the dataset undergoes refinement through sampling and feature extraction processes, aiming to encapsulate solely the most pertinent information essential for model training. The ultimate step involves normalization, an imperative process aimed at ensuring data consistency across diverse ranges and scales. Via this series of detailed and thorough preprocessing steps, we have laid a solid groundwork for subsequent machine learning model training, guaranteeing that the data is in an optimal state prior to being fed into the MobileNetV3 model.

1) CAR-HACKING DATASET

Within this study, the dataset employed for internal vehicle analysis is the Car-Hacking dataset, curated by capturing CAN traffic through the OBD-II port during occurrences of CAN attacks [29]. This dataset is tailored specifically for the exploration of vehicle network security. It provides data extracted from the Controller Area Network (CAN) bus of contemporary automobiles or alternative vehicle communication interfaces. The Car-Hacking dataset encompasses a spectrum of attacks on vehicular networks, including but not limited to: Distributed Denial of Service (DoS) attacks, fuzzing attacks, and deception attacks [30]. The plethora of attack types embodied in the dataset renders it an invaluable resource for researching vehicular network security, especially in the development of advanced systems tailored for the detection and mitigation of such attacks.

The primary features of the Car-Hacking dataset include CAN identifiers (IDs) and the 8-bit data fields of CAN traffic packets (DATA[0]-DATA [7]) [31], [32]. The meticulous documentation of these characteristics not only unveils the attributes of the attack process but also furnishes critical data for analyzing patterns of aggressive behavior. The dataset encompasses five labeled categories: 'Normal', 'RAM', 'Gear ', 'DoS ', and 'Fuzzy'. The distribution of each labeled dataset is illustrated in Table 1 as follows. Evidently, during the normal operation of vehicles, the quantity of normal data significantly surpasses that of anomalous data. Furthermore, given the vast amount of data, the inevitable presence of some outliers is unavoidable. Therefore, in the data preprocessing phase, we implemented filtering and cleaning of the data to avoid biases during the training process, which could lead to overfitting phenomena.

TABLE 1. **Distribution of the Car-Hacking label dataset.**

Labels	Quantity
Normal	701,832
Fuzzy	24,624
DoS	29,501
Gear	29,944
RAM	32,539

2) CICIDS-2017 DATASET

For external network intrusions targeting vehicles, we employ the CICIDS-2017 dataset, a network traffic dataset developed by the Canadian Institute for Cybersecurity and Intelligent Cities Laboratory (CIC). The dataset includes a variety of real-world attack types such as DDoS (Distributed Denial of Service), DoS (Denial of Service), Web attacks, and information leaks. The CICIDS-2017 dataset encompasses diverse network traffic characteristics, including source IP, destination IP, transport layer protocol, flow duration, and the number and size of packets [33], [34], [35]. These features are utilized to train machine learning models to distinguish between normal and anomalous traffic. Although the CICIDS-2017 dataset was initially conceived for conventional network settings, it covers an extensive array of network attack types that are critically pertinent to threats that vehicular communication networks might confront. Moreover, numerous cybersecurity strategies and methodologies devised for Ethernet settings are transferable to alternative arenas, including vehicular networks. Vehicular networks, especially the telematics systems in contemporary vehicles, depend profoundly on the robustness and security of network operations. The cyber attacks emulated in the CICIDS-2017 dataset, such as DoS and DDoS, resemble potential cyber threats these systems could face. By leveraging this dataset, researchers can evaluate the efficacy of extant network intrusion detection systems against vehicular network security threats, and refine and enhance these systems for the unique communication environments of vehicles. This dataset comprises data labeled with 12 different categories, distributed as depicted in Table 2. Analogous to the Car-Hacking dataset, the prevalence of normal data samples substantially outnumbers that of anomalous data, necessitating similar approaches in dataset preprocessing.

TABLE 2. **Distribution of the CICIDS-2017 label dataset.**

Labels		Quantity
BENIGN		2,273,097
DoS Hulk		231,073
PortScan		158,930
DDoS		128,027
DoS GoldenEye		10,293
DoS Slowhttptest		5,499
FTP-Patator		7,938
SSH-Patator		5,897
DoS slowloris		5,796
Web Attack	Web Attack - Brute Force Web Attack - XSS Web Attack - Sql Injection	2,180
Bot		1,966
Infiltration		36
Heartbleed		11

3) DATA FILTERING

Initially, we introduced an intrusion detection system (IDS) paradigm that commenced processing the dataset. Specifically for the Car-Hacking dataset, given the substantially greater volume of normative data compared to anomalous data within the automotive network data, we implemented filtration and stratification strategies. We preserved the quantities of samples labeled with 'Fuzzy', 'RPM', 'Gear', 'DoS', and analogous designations to ensure these pivotal anomaly types were adequately represented. Furthermore, to tackle the problem of data imbalance, we executed a 50% without-replacement stochastic sampling of the normative data samples marked 'Normal'. In the instance of the CICIDS-2017 dataset, pronounced imbalance was discernible among the datasets. Hence, we conducted stochastic sampling of the 'BENIGN', 'DoS Hulk', 'PortScan', 'DDoS' designations according to their respective proportions, whereas the remainder of the dataset remained unaltered. This approach aimed to equilibrate the proportion of samples within the dataset, thereby augmenting the efficacy and precision of model training. Following these procedures, we amalgamated the sampled data with the unsampled samples and arranged them by their indices. This process ensured the coherence and integrity of the dataset.

4) ISOLATED FOREST OUTLIER PROCESSING

Artificial intelligence algorithms frequently entail the manipulation of extensive datasets that contain both pertinent and superfluous data. Given the occurrence of anomalies within these datasets, the Isolation Forest technique is utilized for anomaly mitigation. Should an aberrant field be identified, it is consequently eliminated. This technique, being unsupervised, necessitates the exclusion of the label column. The Isolation Forest is executed by arbitrarily selecting an attribute and a random division value for that attribute, with the division rule of the stochastic decision trees formulable as follows.

$$f(x) = \begin{cases} q, & \textit{if } x_q < p \\ \neg q, & \textit{otherwise} \end{cases} \tag{1}$$

Herein, q signifies the attribute, p denotes the partition point, and x_q is the valuation of data point x with respect to attribute q.

Regarding the calculation of anomaly scores, the more rapid isolation of a data point (namely, the more abbreviated the path length), the elevated the anomaly score becomes. Isolation forests ascertain the anomaly score based on the mean path length of the data points, as depicted in the subsequent equation.

$$S(x, n) = 2^{-\frac{E(h(x))}{C(n)}} \tag{2}$$

$$C(n) = 2 \cdot H(n-1) - \frac{2(n-1)}{n} \tag{3}$$

Among these terms: E(h(x)) signifies the mean path length for the data point x, C(n) represents the anticipated value of the mean path length, and H(i) is designated as the harmonic number, delineated as H(i) = ln(i) + 0.5772156649. In this context, n refers to the total sample count within the dataset, and the path length h(x) describes the sequence of edges encountered from the root node to the leaf node where x resides.

In deploying the Isolation Forest algorithm, we orchestrated a framework consisting of 100 decision trees to forge a model architecture tailored for anomaly detection. Concurrently, the contamination factor was established at 0.1, implying that an estimated 10% of the data points within the dataset are projected to be anomalies. Furthermore, to guarantee the reliability and replicability of our experimental findings, a constant random seed was specified.

With respect to the Car-Hacking dataset, the outcomes of anomaly detection are illustrated in Figure 4. Initially, features DATA [5] and DATA [6] demonstrated pronounced negative correlations, with coefficients of −0.51 and −0.53 respectively. This suggests that elevated values of these attributes significantly increase the likelihood of the data points being classified as anomalous by the model. The CAN ID presented a negative correlation coefficient of −0.24, which, although more subdued, still indicates a certain proficiency in detecting anomalies when CAN ID values are elevated. This might suggest that CAN ID could serve as a supportive characteristic in identifying anomalous conditions. Other attributes like DATA[0], DATA [2], and DATA [3] exhibited correlation coefficients ranging from −0.42 to −0.37, indicating mild to moderate negative correlations. Features DATA [1], DATA [4], and DATA [7], with correlation coefficients close to zero at 0.20, 0.34, and −0.10, respectively, suggest a negligible impact in anomaly detection.

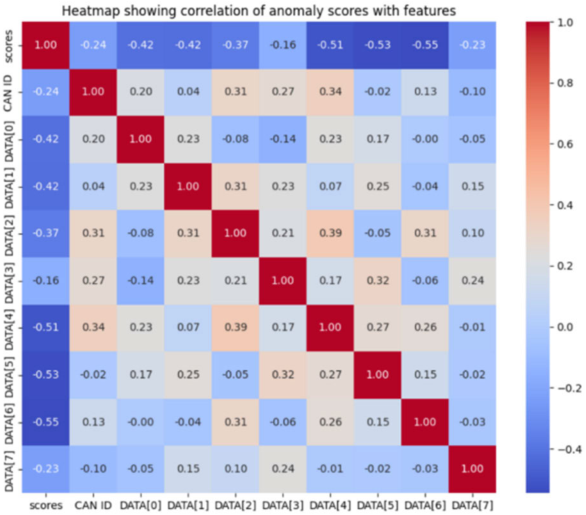


FIGURE 4. Results of anomaly detection in the Car-Hacking dataset.

For the CICIDS-2017 dataset, owing to its considerable assortment of feature columns, it was subjected to dimensionality reduction via Principal Component Analysis (PCA). The detection outcomes are exhibited in Figure 5, where the gradation of colors denotes the anomaly scores associated with the data points. As delineated by the adjacent color legend, hues approaching red reflect elevated anomaly scores, whereas those nearing blue represent diminished scores. Consequently, after an exhaustive assessment of all feature columns for anomalies, data points deemed outliers based on the predictive labels of the model were excised from the dataset. This phase is pivotal for purifying the data and bolstering the precision of the experiments. This preprocessing of data secures the efficacy of model training and subsequent analytical evaluations, furnishing a more robust and dependable groundwork for the comprehensive analysis and elucidation of the data.

5) SMOTE OVERSAMPLING

In addressing the problem of outliers, we eliminated a multitude of data points that were extraneous to the experiment, which led to a disparity in the distribution of category labels across the dataset. To counteract the likelihood of overfitting within our intrusion detection model and to augment the precision of the Intrusion Detection System (IDS), we adopted a strategy of oversampling to equilibrate the data. We specifically implemented the SMOTE algorithm, as delineated by the ensuing formula. Within the Car-Hacking dataset, a considerable variance was noted in the number of samples labeled ‘Fuzzy’ relative to other categories. To rectify this imbalance, we applied oversampling to the ‘Fuzzy’ labeled data, raising its count to align with the mean sample sizes of the ‘RPM’, ‘Gear’, and ‘DoS’ categories. For the CICIDS-2017 dataset, we similarly escalated the numbers of ‘Web Attack’, ‘Infiltration’, ‘Bot’, ‘DoS slowloris’, ‘SSH-Patator’, ‘DoS GoldenEye’, ‘FTP-Patator’, and ‘DoS Slowhttptest’ to correspond with the average figures of ‘PortScan’, ‘DoS Hulk’, and ‘DDoS’. This methodology not only harmonizes the proportions of categories but also amplifies the model’s capacity to generalize, thus enhancing its robustness in the face of novel attacks. Following this resampling, an equilibrium was achieved across the datasets, ensuring that the IDS sustains elevated levels of detection accuracy in diverse contexts.

$$x_{new} = x_i + rand(0, 1) \times (x_{nn} - x_i) \tag{4}$$

Herein, x_i represents each sample from the minority class, x_{nn} denotes the nearest neighbor sample, x_{new} symbolizes the newly synthesized sample, and rand(0,1) is a random number drawn from a uniform distribution.

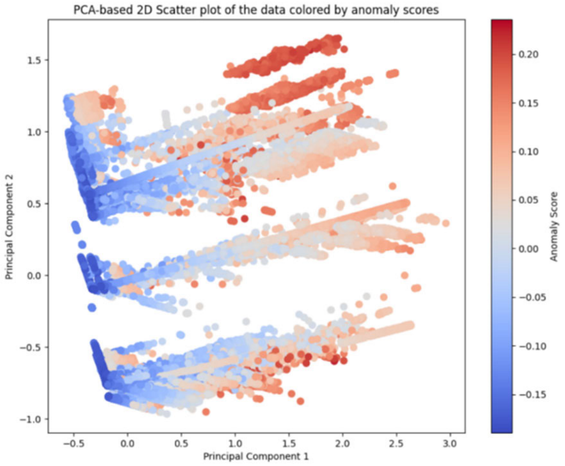


FIGURE 5. Results of anomaly detection in the CICIDS-2017 dataset.

6) RANDOM FOREST FEATURE SELECTION

To augment the detection efficiency and precision of the model, feature selection is implemented on the dataset, isolating attributes whose importance scores surpass a predetermined threshold. These attributes are considered pivotal to the model’s predictive efficacy. This method facilitates the identification and preservation of the most critical features for the prediction endeavor, whilst discarding those of lesser significance. The model utilizes a random forest classifier and conducts feature selection through the computation of Gini impurity, which quantifies the importance of features within the trees, as delineated by the subsequent formula.

$$Imp(X_j, T) = \sum\nolimits_{t \in T} \Delta I(t, X_j) \tag{5}$$

Within this framework, $\Delta \mathrm{I}(\mathrm{t}, \mathrm{X_j})$ represents the decrement in impurity of feature X_j before and after the split at node t. Subsequently, by summing across all trees and computing the average, one obtains the aggregate importance of feature X_j:

$$Imp(X_j) = \frac{1}{M} \sum\nolimits_{i=1}^{M} Imp(X_j, T_i) \tag{6}$$

Features from the Car-Hacking dataset are selected predicated on their significance, preserving only those with a significance value greater than 0.05, as delineated in Figure 6.

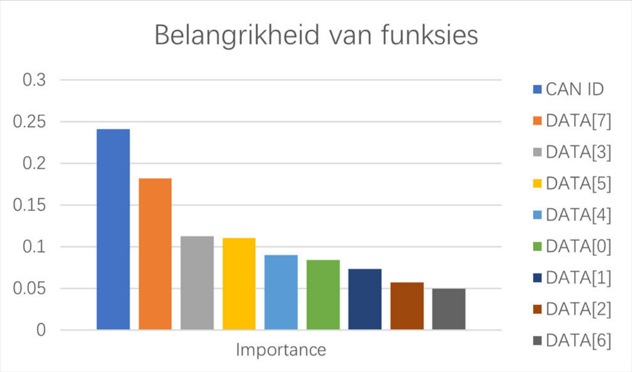


FIGURE 6. The quantitative assessment of feature significance within the Car-Hacking dataset.

7) TRANSFORM THE IMAGE

In the realm of image manipulation, normalization constitutes an essential pre-processing step that enhances the performance of algorithms by standardizing the range of pixel values. Typically, pixel values span the [0,255] range; however, employing a quantile-based normalization method proves to be an effective means of data standardization. This method is particularly adept at recalibrating the data to be more centrally aligned with the median, assuming a Gaussian distribution. Consequently, it diminishes the influence of outliers and bolsters the robustness of the dataset. The mathematical formulation of the quantile transformation is delineated below.

$$x'_{ij} = \frac{F_j(x_{ij}) - F_j(q_l)}{F_j(q_u) - F_j(q_l)} \tag{7}$$

Within this schema, each feature within the dataset is characterized by its cumulative distribution function (CDF). The expressions q_l and q_u signify the lower and upper quantiles, respectively, while $\mathrm{F_j(q_l)}$ and $\mathrm{F(q_u)}$ represent the values of the cumulative distribution function associated with these quantiles, respectively.

In the course of preparing the training data for the MobileNetV3 model, we executed a series of meticulous data preprocessing protocols. These encompassed data sampling, outlier rectification, normalization, and feature enhancement. Subsequently, the network traffic data samples from our dataset were converted into images of 224 × 224×3 format. This image configurations effectively harnesses the

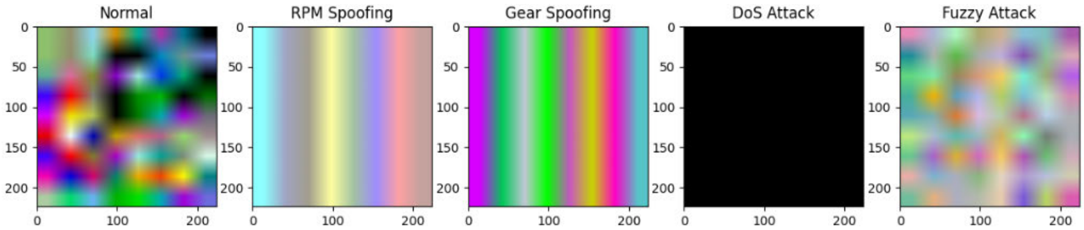


FIGURE 7. Representative image samples of various attack types in Car-Hacking datasets.

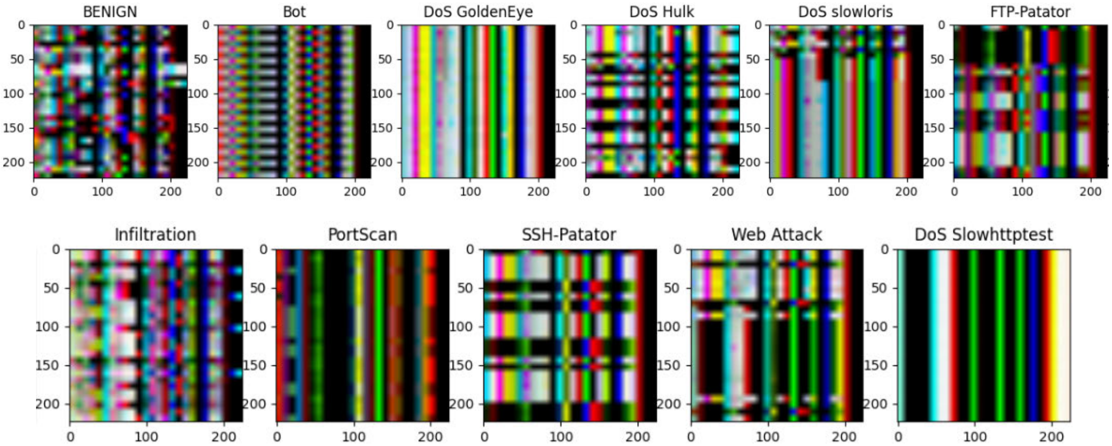


FIGURE 8. Representative image samples of various attack types in CICIDS-2017 datasets.

architectural features of MobileNetV3, ensuring appropriate input. Illustrative image samples representing various types of attacks from the Car-Hacking dataset are depicted in Figure 7, whereas Figure 8 displays similar samples from the CICIDS-2017 dataset. Through the examination of these detailed image attributes, the MobileNetV3 model achieves heightened precision in identifying and distinguishing among various network attacks, thereby serving a crucial role in security defense in real-world applications.

C. THE PROPOSED INTRUSION DETECTION MODEL

In the pivotal phase of data preprocessing, all images undergo a transformation to conform to the 224 × 224×3 standard, a requirement tailored to align with the input criteria of the MobileNetV3 model. This critical formatting endeavor is indispensable for the model's efficacy as it guarantees data uniformity and facilitates efficient information extraction from the images. Following this transformation, the preprocessed images are introduced into the intricately engineered MobileNetV3 model for comprehensive training and validation. Celebrated for its computational efficiency and precision, MobileNetV3 is exceptionally adept for applications on mobile and edge devices. Within the model's framework, depicted in Figure 9, the initial phase concentrates on spatial feature extraction through a synergistic operation of residual connections and dual convolutional layers. Subsequent to each convolutional layer, batch normalization and ReLU activation functions are employed to enhance the capture of spatial patterns at the early model stages. These initial features are then refined in successive stages through a sequence of inverted residual blocks, facilitating intricate feature transformations. The strategic use of residual connections not only amplifies the network's training efficiency but also bolsters the model's ability to generalize. Following the intricate processing by the inverted residual blocks, the model advances to the classification phase. This final stage employs global average pooling, additional convolutional layers, and fully connected layers, culminating in the generation of classification outcomes via the output layer. This carefully orchestrated structural amalgam endows MobileNetV3 with the dual benefits of lightness and formidable processing capacity and accuracy.

In the reverse residual block of MobileNetV3, an intricately configurations sequence of strata is utilized to enhance network efficacy: an expansion layer, a depthwise separable convolution layer, an attention mechanism layer, and a projection layer. The intricate operational flow of these strata, along with the internal functional management, is depicted in Figures 10 and 11. Notably, an initial evaluation of the input and output channel counts is conducted to ascertain their equivalence. Should they coincide, the expansion layer is bypassed, and the input is seamlessly transitioned to the depthwise separable convolution layer for refinement.

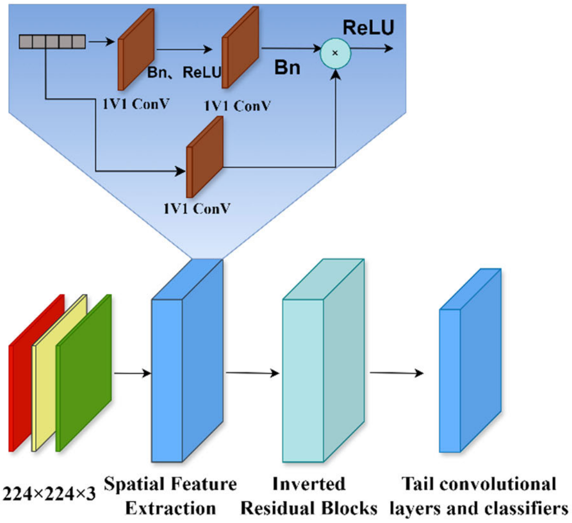


FIGURE 9. The network architecture of MobileNetV3 model.

Contrarily, if a disparity exists, the input undergoes modulation in the expansion layer to align the channel counts, thus ensuring parity between the input and output channel numbers. The expansion layer employs 1 × 1 convolutional kernels and ReLU6 activation functions initially to amplify the channel count of feature maps. This strategy is designed to enrich the detailed feature information available for the ensuing depthwise separable convolution layer while maintaining the dimensional congruence of the block's input and output.

The depthwise separable convolution layer, an efficacious strategy in convolution operations, substantially reduces both the parameter count and computational overhead while sustaining superior performance. This method is eminently suitable for streamlined network designs and deployment on mobile platforms. The procedure is bifurcated into two integral stages: depthwise convolution and pointwise convolution. Initially, depthwise convolution is implemented, subsequently followed by pointwise convolution. During depthwise convolution, designated Bottlenecks execute convolution operations independently on each channel of the input feature map, utilizing individual filters per channel. This approach primarily facilitates the extraction of features within singular channels, rather than across all channels concurrently. The Bottleneck configurations might include a single filter, the application of attention mechanisms, types of activation functions, and stride specifications. Following the completion of depthwise convolution, the phase of pointwise convolution initiates. This stage employs 1 × 1 filters to manipulate the output from the depthwise convolution. In contrast to depthwise convolution that focuses on intra-channel feature extraction, pointwise convolution aims to amalgamate features inter-channel. Through linear combinations of features across various channels, this stage effectively consolidates the spatial attributes extracted during depthwise convolution into a unified feature representation, thereby enhancing the model's expressive power. Within the entire inverted residual block structure, if both the input and output feature map dimensions remain unchanged and the stride for depthwise separable convolution is set at 1, a skip connection strategy is implemented; if not, the utilization of skip connections is precluded.

The merit of this staged processing technique resides in its pronounced reduction of parameter quantities and computational demands. In depthwise convolution, the application of filters individually within each channel substantially curtails the parameter count; following this, the pointwise convolution effectuates the amalgamation of features across various channels whereas preserving computational efficiency. This confluence of procedures establishes depthwise separable convolution as both efficacious and formidable, particularly apt for environments with limited resources. Throughout this process, the deployment of activation functions (such as ReLU or Hardswish) and, when employed, attention mechanisms, significantly bolsters the network's capability for nonlinear processing and the judicious distribution of feature importance. Additionally, each convolutional stage may incorporate normalization techniques, like batch normalization, which contribute to the training stability and efficiency of the network.

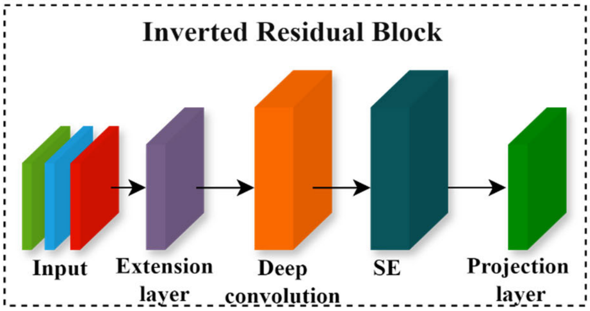


FIGURE 10. Inverted residual network of MobileNetV3.

The deployment of the Squeeze-and-Excitation (SE) attention architecture transpires subsequent to the deep convolutional layers, facilitated by configurations variables within the inverted residual structures. The quintessential role of the SE attention framework is to dynamically modulate the weight distribution across diverse regions of the feature maps, thereby amplifying the model's proficiency in apprehending pivotal information. Its architecture is depicted in Figure 12. This framework encompasses two crucial constituents: Squeeze and Excitation. Initiated by the Squeeze component, the process commences with average pooling applied to the input feature maps, effectuating spatial condensation of each channel to formulate a comprehensive feature representation. Thereafter, the Excitation component assimilates these condensed features and subjects them to two sequential convolutional layers. The initial convolutional stratum utilizes a 1 × 1 convolutional kernel in conjunction with the ReLU activation mechanism, serving to diminish the parameter count while integrating non-linearity;

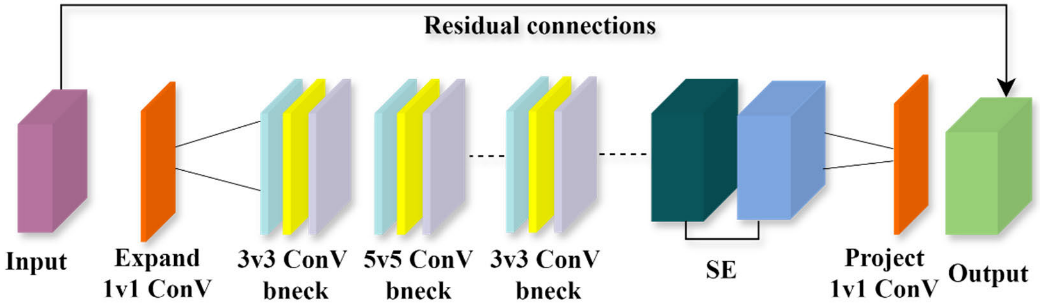


FIGURE 11. Internal function processing of MobileNetV3 reverse residual block.

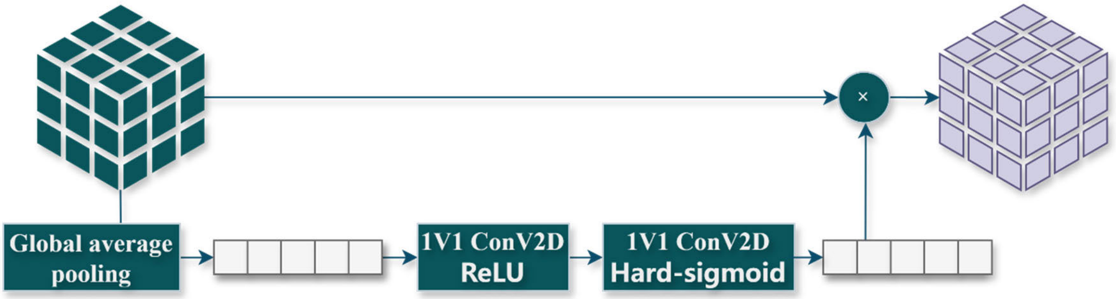


FIGURE 12. The execution process of the SE attention mechanism.

the succeeding convolutional stratum similarly employs a 1 × 1 kernel, it is combined with a Hard-sigmoid activation mechanism, intended to produce attention weights for each channel.

Employing this sophisticated architecture, the SE attention mechanism meticulously calculates the importance coefficients for each channel. These coefficients are subsequently employed to modulate the original feature map channels, thereby recalibrating the feature maps to accentuate more critical channels and attenuate the less significant ones. This refined feature representation ensures that the model's output is more attuned to information that is essential for the definitive task, significantly enhancing the model's overall efficacy and efficiency.

In conclusion, the inverted residual architecture typically incorporates a projection layer designed to recalibrate the feature map dimensions to conform to the desired output specifications. This layer employs a 1 × 1 convolutional operation, adjusting the channel count to satisfy the model's output criteria. Subsequent to the phase of feature extraction, the architecture applies a global average pooling to the feature maps via an adaptive average pooling layer, compressing the spatial dimensions to a singular unit. Classification is then executed through a classifier, which comprises a linear fully connected layer coupled with an activation function. A Dropout layer precedes the ultimate fully connected layer to curtail overfitting. The output from this final layer is processed through a Softmax function, resulting in a probability distribution across the categories. The model identifies and selects the category with the highest likelihood as the definitive classification outcome.

IV. EXPERIMENTAL OUTCOMES AND PERFORMANCE EVALUATION

In our study, the programming was conducted using the Python language, employing a suite of specialized libraries including TensorFlow, NumPy, Hyperopt, Matplotlib, Scikit-learn, and Pandas. These libraries offer a comprehensive range of functionalities that support various stages from data processing and model training to results visualization. Moreover, our experiments utilized an Intel XEON Gold 6226R CPU, accompanied by 32GB of memory, an NVIDIA RTX 3090 GPU, and a 1.92TB SATA solid-state drive (SSD). Such hardware configurations provided robust support for our research, ensuring efficiency and stability in computational tasks.

A. ASSESSMENT CRITERIA

To ascertain the interoperability between the proposed Intrusion Detection System (IDS) framework and the MobileNetV3 architecture, and to obviate performance detriments stemming from unsuitable hyperparameter tuning, we utilized the Tree-structured Parzen Estimator (TPE) for the refinement of hyperparameters. TPE stands out as an efficacious strategy for the optimization of hyperparameters, simultaneously minimizing computational expenditures and safeguarding precision. We delineated a hyperparameter exploration domain encompassing pivotal parameters such as batch size, iteration frequency, and learning rate. In the realm of data management, to avert the risk of overfitting within our experimental outcomes, we segmented the training dataset following an 80%-20% distribution, thereby constituting an 80% training subset and a 20% testing subset. Furthermore, we established a threshold of five iterations, termed 'patience', as a parameter for prematurely ceasing training. Within the TPE methodology, the election of hyperparameters pivots on the computation of probability ratios. This algorithm meticulously evaluates the efficacy of diverse hyperparameter amalgamations and revises the probabilistic model in light of these evaluations, thus steering the subsequent hyperparameter explorations. This approach allows TPE to adeptly pinpoint and select the most advantageous hyperparameter configurations, thereby augmenting the model's comprehensive efficacy. The formula for calculating the probability ratio in the TPE framework is presented herewith.

$$r(x) = \frac{P(x|y = L)}{P(x|y = H)} \tag{8}$$

Herein, r(x) denotes the relative likelihood of a new configuration x between good and poor configurations, characterized by the PDF of the good configuration ($P(x \mid y = L)$, where LL signifies lower loss) and the PDF of the poor configuration ($P(x \mid y = H)$, where H indicates higher loss).

Upon securing the optimal hyperparameter ensemble, to circumvent the perils of overfitting and guarantee that the model refined through hyperparameter optimization manifests commendable generalizability across diverse data distributions, quintuple cross-validation is leveraged to assess the efficacy of the optimal hyperparameter configuration. In each cycle of this process, 80% of the pristine training dataset is allocated for model training, whereas the residual 20% serves the purpose of model validation. The evaluation outcomes are quantitatively assessed utilizing metrics such as accuracy, precision, recall, and F1 score, with their respective computation methods delineated subsequently.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{9}$$

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{12}$$

In this context, TP signifies the number of true positives, i.e., cases that are positive and accurately identified as such. FN represents false negatives, instances in which positive cases are incorrectly labeled as negative. FP pertains to false positives, denoting instances where negatives are wrongly classified as positives, while TN refers to true negatives, the correct identification of negative instances. Accuracy, precision, recall, and the F1 score are pivotal indices for evaluating the efficacy of classification models, mirroring the predictive strength of a model across varied dimensions. Accuracy is the ratio of correct predictions (regardless of class) to the total predictions made, ideally suited to datasets with uniform distributions. Precision, however, concentrates on the ratio of true positives within the subset predicted as positive. This index is especially vital in environments where the consequences of erroneously classifying a negative as a positive are substantial. In contrast, recall assesses the ratio of actual positives that are correctly detected by the model, essential in situations where failing to detect a positive has significant repercussions. The F1 score, a harmonic mean of precision and recall, integrates both measures and is particularly advantageous for datasets with skewed distributions. Together, these indices aim to thoroughly assess model performance, ensuring the model's effectiveness in fulfilling specific operational requirements. They provide a comprehensive, multi-dimensional framework for performance evaluation, facilitating a more profound understanding and enhancement of classification models.

B. RESULTS AND DISCUSSION

This research endeavors to devise an intrusion detection system characterized by its high efficiency, precision, lightweight structure, and ease of deployment, specifically designed to meet the real-time demands of vehicles and to shield their operation from a variety of malicious traffic intrusions. Accordingly, model training and evaluation were conducted using the Car-Hacking and CICIDS-2017 datasets, with a subsequent multifaceted analysis of the experimental outcomes.

Initially, concerning the dataset pertaining to internal vehicle attacks, after the exclusion of outliers, it was observed that there was a disparity in data volume between the datasets (with the 'Normal' and 'Fuzzy' labels undergoing outlier removal, and 'Fuzzy' category containing more outliers, whereas 'Normal' less). To rectify this imbalance and to augment the precision of the intrusion detection system, the Synthetic Minority Over-sampling Technique (SMOTE) was utilized to equilibrate the distribution of classes within the dataset. The distribution of classes after this adjustment, as illustrated in Table 3, not only enhanced the quality of data but also facilitated the model's capability and accuracy in processing imbalanced data. Through this methodology, this research aspires to boost the overall functionality of the intrusion detection system, ensuring its capability to effectively identify and counteract diverse network threats, thus securing the continuous safe operation of vehicles.

In the realm of the CICIDS-2017 dataset, originally formulated for the identification of network intrusions and irregularities, its encompassing array of attack scenarios combined with genuine traffic attributes render it suitable for integration into our external vehicle intrusion detection frameworks. Following an initial phase of outlier curation, data categorized under the 'Heartbleed' label type was entirely expunged, along with a reduction observed in other labeled datasets. Subsequently, we employed the SMOTE (Synthetic Minority

TABLE 3. Distribution information of Categories after Processing the Car-Hacking dataset.

Category	Before outlier processing	After outlier processing	After SMOTE oversampling
Normal	350,916	328,551	328,551
RAM	32,539	32,539	32,539
Gear	29,944	29,944	29,944
DoS	29,501	29,501	29,501
Fuzzy	24,624	260	30,661

Over-sampling Technique) algorithm to preprocess the data, with the objective of harmonizing the distribution of diverse categories within the dataset. The distribution of categories post-preprocessing is delineated in Table 4 (owing to the extensive array of data categories, in this table, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowloris, DoS Slowhttptest, and Heartbleed have been amalgamated under the DoS label data; similarly, SSH-Patator and FTP-Patator have been consolidated into the Brute-Force label data).

TABLE 4. Distribution information of categories after processing the CICIDS-2017 dataset.

Category	Before outlier processing	After outlier processing	After SMOTE oversampling
BENIGN	409,157	384,932	384,932
DoS	99,821	68,259	170,777
PortScan	31,786	31,763	31,763
Brute-Force	13,835	13,835	57,868
Bot	1,966	1,922	28,934
Web Attack	2,180	2,092	28,934
Infiltration	36	7	28,934

Owing to the substantial discrepancies between the Car-Hacking dataset and the CICIDS-2017 dataset, our approach involved independently training and evaluating models tailored to each dataset. Following a meticulous optimization of hyperparameters for these datasets, we ascertained an exemplary configuration of hyperparameters, delineated in Tables 5 and 6. Thereafter, to validate the efficacy of these hyperparameters, we implemented a quintuple cross-validation technique within our model framework. Throughout the quintuple cross-validation process applied to the Car-Hacking dataset, the model consistently exhibited stellar performance, achieving perfection with 100% scores across accuracy, precision, recall, and F1 metrics. Similarly, for the CICIDS-2017 dataset, the metrics of accuracy, precision, recall, and F1 all surpassed 99.40%. These outcomes signify that the model's performance on the test datasets during the quintuple cross-validation reached an optimal condition. Collectively, these findings robustly demonstrate our model's effectiveness and precision in processing distinct datasets, particularly after extensive hyperparameter tuning and repeated validations. The attained precision in predictive capabilities underscores the model's robustness and validity in pertinent computational fields.

TABLE 5. Optimal hyperparameter configuration for the Car-Hacking dataset.

Variable	Value
Batch_size	16
Epochs	5
Learing rate	0.0010688470553860802

TABLE 6. Optimal hyperparameter configuration for the CICIDS-2017 dataset.

Variable	Value
Batch_size	4
Epochs	5
Learing rate	0.0012866150092685834

Upon concluding quintuple cross-validation, we performed a definitive performance evaluation of the model utilizing optimally adjusted hyperparameters on a designated test dataset. To avert overfitting, this dataset was not previously employed in any training evaluations. Pertaining to the Car-Hacking dataset, the evaluation outcomes regarding accuracy, loss rates, and confusion matrices are illustrated in Figures 13 and 14. The confusion matrix reveals an almost exemplary classification outcome, with each category (Dos, Fuzzy, Normal, RAM, Gear) exhibiting substantial true positive values, indicating accurate classifications with virtually no misclassifications. In the loss and accuracy charts, we witness a precipitous reduction in test loss to nearly zero after a single batch, signifying the model's exemplary performance on the test data with minimal loss. Concurrently, the accuracy chart portrays a consistently high accuracy level, remaining close to 100% throughout the testing phase, thereby affirming the model's enduring precision.

Similarly, for the CICIDS-2017 dataset, the assessment of accuracy, loss rates, and confusion matrices as delineated in Figures 15 and 16, demonstrates the model's classification effectiveness across various categories (represented from classes 0 to 11). Despite classification inaccuracies in categories 3 and 6, the results for the remaining categories were remarkably superior. Following the initial batches, the loss trajectory sharply descended, rapidly approaching zero, indicating that the model adeptly generalized the test data with exceedingly low error rates. The accuracy trajectory shows a stable performance, with accuracy levels consistently near 100% across most batches, exhibiting minimal deviations. This high accuracy, complemented by high true positive rates in the confusion matrix, collectively corroborates the model's efficacious processing of the CICIDS-2017 dataset.

Focusing on the Car-Hacking dataset, to underscore the significance of the proposed research model, this study not only utilized the introduced model but also juxtaposed it against a range of foundational deep learning models,

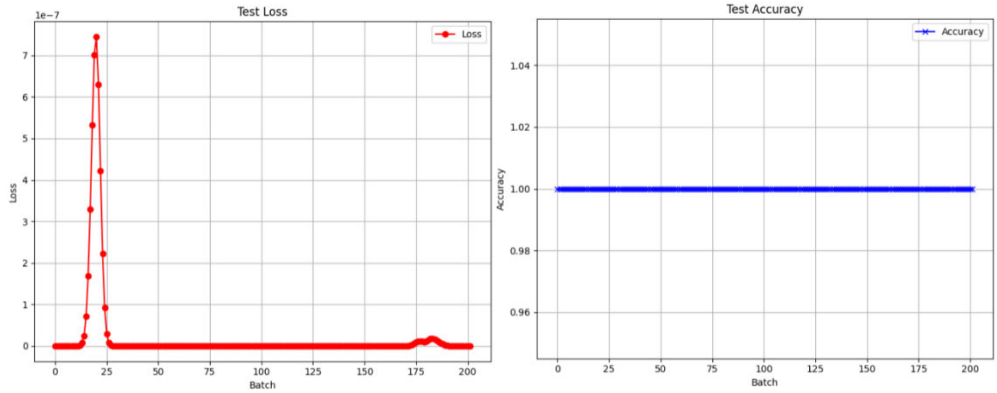


FIGURE 13. Accuracy and loss metrics of Car-Hacking dataset evaluations.

as delineated in literature [29], [36], [37], [38]. These encompass P-LeNet, DCNN, DivaCAN, LSTM, Autoencoder, Concatenated CNN, and InceptionResnet-PSO. The performance evaluation of these models on the Car-Hacking dataset, encompassing pivotal metrics such as accuracy, precision, recall, F1 score, and test time per packet, is documented in Table 7. The evaluation highlights that the MobileNetV3 model exhibited significant superiority across all evaluation metrics in comparison to conventional deep learning models such as P-LeNet, DCNN, and DivaCAN. Notably, precision surged dramatically from 85.89% to 100% compared to the DivaCAN model, a substantial enhancement of 14.11%. The memory footprint was 16MB. Furthermore, compared to sequence-based models like LSTM and Autoencoder, especially the LSTM model, which otherwise showcased commendable performance, the F1 score could be elevated from 96.82% to 100% with MobileNetV3, showcasing a stronger generalization capability.

TABLE 7. Evaluation results of the model on the Car-Hacking dataset.

Method	Accuracy(%)	Precision(%)	Recall(%)	F1(%)	Test Time PerPacket (ms)
P-LeNet	98.10	98.14	98.04	97.83	-
DCNN	99.93	99.84	99.84	99.91	-
DivaCAN	94.93	85.89	94.98	94.97	-
LSTM	96.03	96.18	96.17	96.82	
Autoencoder	99.98	99.96	99.85	99.96	
Concatenated CNN	100	100	100	100	3.2
InceptionResnet-PSO	100	100	100	100	1.3
MobileNetV3	100	100	100	100	1.0

Particularly noteworthy is MobileNetV3's performance in terms of test timing, clocking at merely 1 millisecond per packet. This not only indicates superior efficacy but also significantly surpasses other models in efficiency. For instance, although the Concatenated CNN also achieved 100% across several key performance metrics, its test timing was 3.2 milliseconds—threefold that of MobileNetV3. Even compared to InceptionResnet-PSO, which exhibited commendable performance across various indicators, its test timing of 1.3 milliseconds remained inferior to MobileNetV3.

Examining the specifics of the Car-Hacking dataset further highlights the proposed intrusion detection model's advantages. As shown in Table 8, our experiments, utilizing a reduced volume of data, accomplished flawless evaluation results across all classification tasks, demonstrating exceptional performance, which meets the demands for vehicular lightweighting and efficiency. Specifically, the model achieved an ideal state of 1.00 in precision, recall, and F1 score across categories DoS, Fuzzy, Normal, RAM, and Gear. This result indicates that out of 3222 sample data, the model accurately identified and classified every case without exception. Moreover, the overall model's test results on these 3222 samples demonstrated a perfect accuracy, macro-average, and weighted average of 1.00, or 100% accuracy rate, meaning that the model's predictions were impeccably accurate across all categories, showcasing high classification efficacy. This performance underscores the proposed MobileNetV3 intrusion detection network framework's efficiency and reliability in handling large-scale datasets.

TABLE 8. Evaluation results for each label in the Car-Hacking dataset.

	Precision	Recall	F1-score	Support
DoS	1.00	1.00	1.00	230
FuzzY	1.00	1.00	1.00	214
Normal	1.00	1.00	1.00	2318
RAM	1.00	1.00	1.00	230
Gear	1.00	1.00	1.00	230
Accuracy			1.00	3222
Macro avg	1.00	1.00	1.00	3222
Weighted avg	1.00	1.00	1.00	3222

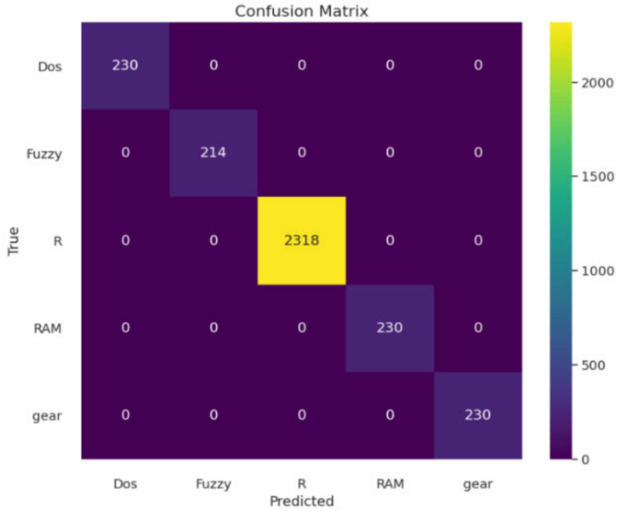


FIGURE 14. Confusion matrix of Car-Hacking dataset evaluation results.

In our exploration of intrusion detection methodologies, we employed an array of machine learning and deep learning algorithms on the CICIDS-2017 dataset. These algorithms, including KNN, DBN, Multi-SVM, PCA-RF, FS XGBoost, STDeepGraph, and MLP, are cited from extant scholarly articles [20], [39], [40], [41], [42], [43], [44]. The assessment outcomes on the CICIDS-2017 dataset are delineated in Table 9. When assessing the efficacy of these algorithms, the MobileNetV3 model surpassed others by achieving the zenith of performance metrics—accuracy, precision, recall, and F1 score—all peaking at an exceptional 99.76%, thereby demonstrating its pronounced superiority. Notably, in comparison with KNN (uniformly scoring 96.3%) and DBN (yielding 98.95%, 95.82%, 95.80%, and 95.81% respectively), MobileNetV3 showcased an enhancement in metrics by roughly 3% to 4%. Moreover, even juxtaposed with the notably effective Multi-SVM (98.55%, 98.22%, 98.38%, and 98.3%), PCA-RF (consistently at 99.6%), FS XGBoost (99.7%, 99.55%, 99.65%, and 99.6%), and MLP (99.46%, 99.52%, 99.4%, and 99.5%)—all demonstrating laudable performances—there were minor variances, such as FS XGBoost's recall slightly lagging behind its precision. The uniformity and marginally superior figures of MobileNetV3 also illustrate its preeminence. Beyond its numerical supremacy, MobileNetV3 is architecturally designed as an efficacious deep learning model, leveraging depthwise separable convolutions to substantially curtail computational demands and parameter counts. This innovative design not only amplifies processing velocity but also diminishes operational expenditures, rendering MobileNetV3 particularly apt for deployment in scenarios that demand swift responsiveness and operational efficiency, such as in mobile and embedded systems. This technological advantage endows MobileNetV3 with enhanced adaptability and elevated utility in practical applications.

TABLE 9. Evaluation results of the model on the CICIDS-2017 dataset.

Method	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)
KNN	96.3	96.2	96.3	96.3
DBN	98.95	95.82	95.80	95.81
Multi-SVM	98.55	98.22	98.38	98.3
PCA-RF	99.6	99.6	99.6	99.6
FS XGBoost	99.7	99.55	99.65	99.6
STDeepGraph	99.4	98.6	99.6	99.1
MLP	99.46	99.52	99.4	99.5
MobileNetV3	99.76	99.76	99.76	99.76

The performance indices for a multitude of network attacks within the CICIDS-2017 dataset are delineated in Table 10, incorporating metrics such as Precision, Recall, F1-score, and the count of instances (Support) for each attack type. Predominantly, the model manifests exceptionally elevated accuracy across a broad spectrum of attack scenarios, achieving the zenith of perfection with scores of 1.00 in Precision, Recall, and F1-score across several categories, namely Bot, DDoS, FTP-Patator, PortScan, SSH-Patator, and Web Attack, underscoring its formidable diagnostic prowess. Distinctly, the model delineates slight variances in managing DoS GoldenEye and DoS Hulk attacks. In the case of DoS GoldenEye, despite a flawless Precision of 1.00, the Recall is marginally diminished at 0.9681, signifying that a few positive instances eluded detection, culminating in an F1-score of 0.9838. In contrast, for the DoS Hulk assault, the Recall achieves a perfect score of 1.00, indicating comprehensive identification of all positive instances, albeit the Precision is marginally reduced to 0.9730, suggestive of the presence of several false positives.

TABLE 10. Evaluation results for each label in the CICIDS-2017 dataset.

	Precision	Recall	F1-score	Support
BENIGN	1.00	0.9996	0.9998	2746
Bot	1.00	1.00	1.00	223
DDoS	1.00	1.00	1.00	168
DoS GoldenEye	1.00	0.9681	0.9838	188
DoS Hulk	0.9730	1.00	0.9863	216
DoS Slowhttptest	0.9860	0.9906	0.9883	213
DoS slowloris	0.9907	0.9861	0.9884	216
FTP-Patator	1.00	1.00	1.00	211
Infiltration	0.9953	1.00	0.9976	212
PortScan	1.00	1.00	1.00	217
SSH-Patator	1.00	1.00	1.00	206
Web Attack	1.00	1.00	1.00	206
Accuracy			0.9976	5022
Macro avg	0.9954	0.9954	0.9954	5022
Weighted avg	0.9976	0.9976	0.9976	5022

An overarching performance evaluation underscores that the model's overall Precision, Recall, and F1-score consistently hover between 0.9954 and 0.9976 across both macro and weighted averages, further validating the model's high efficacy and dependability in the realm of multi-class network attack detection. Additionally, the model's prowess in accurately identifying BENIGN traffic is markedly pronounced,

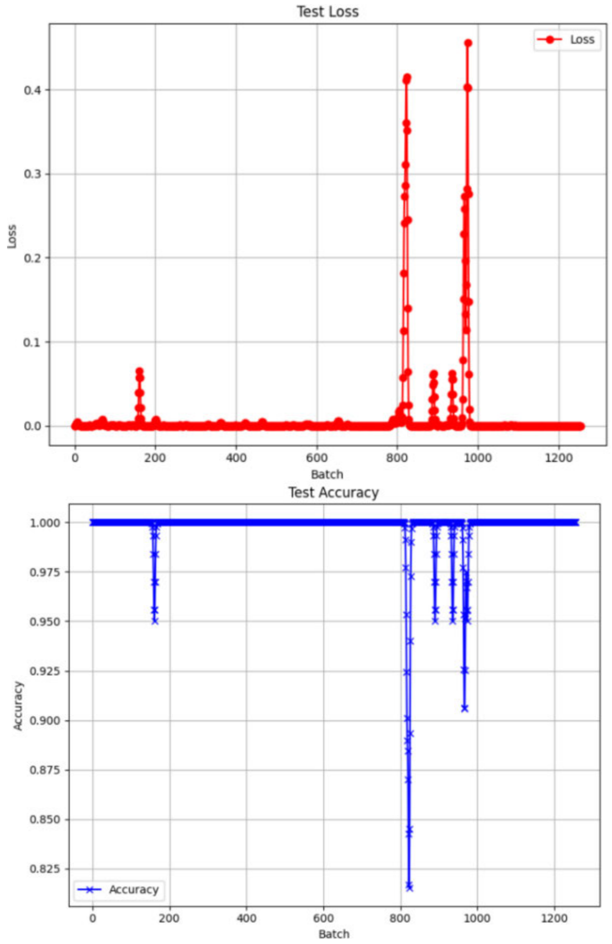
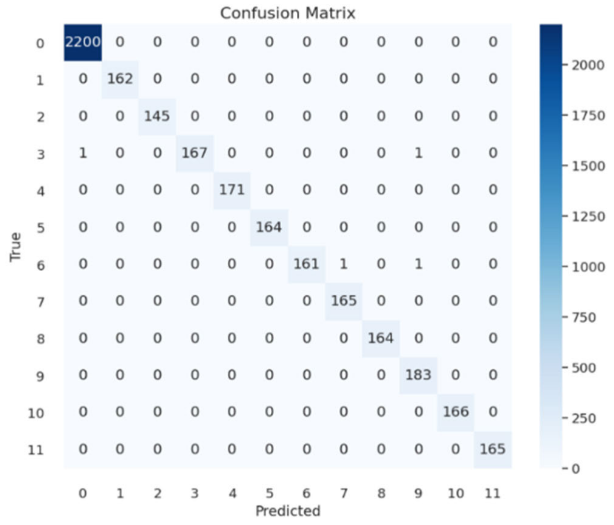


FIGURE 15. Accuracy and loss metrics of CICIDS-2017 dataset evaluations.



Confusion Matrix

True \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11
0	2200	0	0	0	0	0	0	0	0	0	0	0
1	0	162	0	0	0	0	0	0	0	0	0	0
2	0	0	145	0	0	0	0	0	0	0	0	0
3	1	0	0	167	0	0	0	0	0	1	0	0
4	0	0	0	0	171	0	0	0	0	0	0	0
5	0	0	0	0	0	164	0	0	0	0	0	0
6	0	0	0	0	0	0	161	1	0	1	0	0
7	0	0	0	0	0	0	0	165	0	0	0	0
8	0	0	0	0	0	0	0	0	164	0	0	0
9	0	0	0	0	0	0	0	0	0	183	0	0
10	0	0	0	0	0	0	0	0	0	0	166	0
11	0	0	0	0	0	0	0	0	0	0	0	165

FIGURE 16. Confusion matrix of CICIDS-2017 dataset evaluation results.

an essential attribute for minimizing false positives in real-world applications. Collectively, these data not only accentuate the model's exceptional performance across designated attack vectors but also illuminate its strategic value within an integrated network security framework.

Upon meticulous evaluation, the advanced MobileNetV3 architecture not only delivers peak performance in detecting intrusions within and adjacent to vehicles but also offers instantaneous feedback while preserving minimal latency. This is paramount for scenarios that necessitate the rapid identification and management of threats in vehicular networks. Consequently, it is evident that the MobileNetV3 architecture is the superior choice for real-time surveillance in the realm of automotive network security.

V. CONCLUSION

This investigation is dedicated to devising an efficacious, precise, compact, and deployable Intrusion Detection System for vehicular networks, fulfilling the real-time requisites of contemporary vehicles and shielding them from malevolent traffic incursions. Pursuant to this goal, we have introduced an innovative, lightweight detection methodology employing MobileNetV3, subjected to an extensive experimental appraisal. Initially, the research meticulously processed the dataset through techniques such as data filtration, anomaly management, SMOTE oversampling, and feature selection, thereby enhancing data integrity to facilitate model training. The refined data was then input into the MobileNetV3 framework for training and subsequent evaluation. Utilizing a quintuple cross-validation technique, an exhaustive evaluation of the model's performance was executed. The empirical findings affirm that our intrusion detection framework excels in accuracy, recall, precision, and F1 scores on the Car-Hacking and CICIDS-2017 datasets, with a modest footprint of 16MB, demonstrating exceptional efficacy in vehicular cybersecurity. Most notably, the model's testing latency stands at a mere 1 millisecond per packet, vastly surpassing the efficiency of alternative deep learning paradigms, thus highlighting its substantial advantage in operational efficiency. Admittedly, this study has certain limitations. When faced with highly dynamic and covert malicious traffic, our model may exhibit errors or fail to detect threats. Future research could incorporate more advanced deep learning techniques, such as Variational Autoencoders (VAEs) or Generative Adversarial Networks (GANs), to further enhance the model's capability in identifying more complex attack patterns. Moreover, despite our comprehensive data preprocessing steps to ensure data quality and structural suitability, the dataset may still contain biases or incomplete data, potentially impacting the model's generalization ability and practical effectiveness. In real-world environments, data distribution can vary over time and across different scenarios, making it imperative to focus on the model's generalization capacity. Such research will be instrumental in ensuring that intrusion detection systems can effectively address the evolving cybersecurity threats in vehicular networks.

REFERENCES

[1] B. Lampe and W. Meng, "Intrusion detection in the automotive domain: A comprehensive review," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2356–2426, Aug. 2023.

[2] L. Xing, K. Wang, H. Wu, H. Ma, and X. Zhang, "Intrusion detection method for Internet of Vehicles based on parallel analysis of spatio-temporal features," *Sensors*, vol. 23, no. 9, p. 4399, Apr. 2023.

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

[4] L. Silva, N. Magaia, B. Sousa, A. Kobusinska, A. Casimiro, C. X. Mavromoustakis, G. Mastorakis, and V. H. C. de Albuquerque, "Computing paradigms in emerging vehicular environments: A review," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 3, pp. 491–511, Mar. 2021.

[5] B. Sousa, N. Magaia, and S. Silva, "An intelligent intrusion detection system for 5G-enabled Internet of Vehicles," *Electronics*, vol. 12, no. 8, p. 1757, Apr. 2023.

[6] Y. Wang, G. Qin, M. Zou, Y. Liang, G. Wang, K. Wang, Y. Feng, and Z. Zhang, "A lightweight intrusion detection system for Internet of Vehicles based on transfer learning and MobileNetV2 with hyper-parameter optimization," *Multimedia Tools Appl.*, vol. 83, no. 8, pp. 22347–22369, Jun. 2023.

[7] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185489–185502, 2020.

[8] V. Kurama. (2020). *A Review of Popular Deep Learning Architectures: AlexNet, VGG16, and GoogleNet*. Accessed: Jun. 12, 2022. [Online]. Available: https://blog.paperspace.com/popular-deep-learning-architectures-alexnet-vgg-googlenet/

[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[10] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, p. 4396, Oct. 2019.

[11] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019.

[12] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in Internet of Things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, p. 1177, Jul. 2020.

[13] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.

[14] A. Howard and S. Gupta. (2019). *Introducing the Next Generation of On-Device Vision Models: MobileNetV3 and MobileNetEdgeTPU*. [Online]. Available: https://ai.googleblog.com/2019/11/introducing-next-generation-on-device.html

[15] E. Borgia, "The Internet of Things vision: Key features, applications and open issues," *Comput. Commun.*, vol. 54, pp. 1–31, Dec. 2014.

[16] R. G. Engoulou, M. Bellaïche, S. Pierre, and A. Quintero, "VANET security surveys," *Comput. Commun.*, vol. 44, pp. 1–13, May 2014.

[17] J. Greenough. (May 2015). *Connecting Cars to the Internet has Created a Massive New Business Opportunity*. [Online]. Available: http://www.businessinsider.com/connected-car-marketforecast-report2015-5

[18] M. Azees, P. Vijayakumar, and L. J. Deborah, "Comprehensive survey on security services in vehicular ad-hoc networks," *IET Intell. Transp. Syst.*, vol. 10, no. 6, pp. 379–388, 2016.

[19] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 919–933, Mar. 2020.

[20] Y. Yao, L. Su, Z. Lu, and B. Liu, "STDeepGraph: Spatial–temporal deep learning on communication graphs for long-term network attack detection," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Communications/13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 120–127.

[21] K. Aswal, D. C. Dobhal, and H. Pathak, "Comparative analysis of machine learning algorithms for identification of BOT attack on the Internet of Vehicles (IoV)," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 312–317.

[22] D. A. Schmidt, M. S. Khan, and B. T. Bennett, "Spline-based intrusion detection for VANET utilizing knot flow classification," *Internet Technol. Lett.*, vol. 3, no. 3, p. e155, 2020.

[23] T. P. Vuong, G. Loukas, and D. Gan, "Performance evaluation of cyber-physical intrusion detection on a robotic vehicle," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 2106–2113.

[24] W. Fu, X. Xin, P. Guo, and Z. Zhou, "A practical intrusion detection system for Internet of Vehicles," *China Commun.*, vol. 13, no. 10, pp. 263–275, Oct. 2016.

[25] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Secur. (WCICSS)*, Dec. 2015, pp. 45–49.

[26] A. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in *Proc. 5th ESCAR USA Conf.*, 2017, pp. 21–22.

[27] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, "An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks," *IEEE Internet Things J.*, vol. 1, no. 6, pp. 570–577, Dec. 2014.

[28] N. Marchang and R. Datta, "Light-weight trust-based routing protocol for mobile ad hoc networks," *IET Inf. Secur.*, vol. 6, no. 2, pp. 77–83, 2012.

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

[30] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and J.-H. Chen, "Using deep learning networks to identify cyber attacks on intrusion detection for in-vehicle networks," *Electronics*, vol. 11, no. 14, p. 2180, Jul. 2022.

[31] W. Choi, S. Lee, K. Joo, H. J. Jo, and D. H. Lee, "An enhanced method for reverse engineering CAN data payload," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3371–3381, Apr. 2021.

[32] A. Taylor, "Anomaly-based detection of malicious activity in in-vehicle networks," Doctoral dissertation, Ottawa-Carleton Inst. Elect. Comput. Eng., Univ. Ottawa, Ottawa, ON, Canada, 2017.

[33] S. S. Panwar, Y. P. Raiwani, and L. S. Panwar, "An intrusion detection model for CICIDS-2017 dataset using machine learning algorithms," in *Proc. Int. Conf. Adv. Comput., Commun. Mater. (ICACCM)*, Nov. 2022, pp. 1–10.

[34] S. Arshad, W. Ashraf, S. Ashraf, I. Hassan, and F. S. Masoodi, "Comparative study of machine learning techniques for intrusion detection on CICIDS-2017 dataset," in *Proc. 10th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2023, pp. 929–934.

[35] N. Fathima, A. Pramod, Y. Srivastava, and A. M. Thomas, "Two-stage deep stacked autoencoder with shallow learning for network intrusion detection system," 2021, *arXiv:2112.03704*.

[36] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, p. 4736, 2021.

[37] M. H. Khan, A. R. Javed, Z. Iqbal, M. Asim, and A. I. Awad, "DivaCAN: Detecting in-vehicle intrusion attacks on a controller area network using ensemble learning," *Comput. Secur.*, vol. 139, Apr. 2024, Art. no. 103712.

[38] F. W. Alsaade and M. H. Al-Adhaileh, "Cyber attack detection for self-driving vehicle networks using deep autoencoder algorithms," *Sensors*, vol. 23, no. 8, p. 4086, Apr. 2023.

[39] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSp*, vol. 1, 2018, pp. 108–116.

[40] W. Elmasry, A. Akbulut, and A. H. Zaim, "Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic," *Comput. Netw.*, vol. 168, Feb. 2020, Art. no. 107042.

[41] R. Vijayanand, D. Devaraj, and B. Kannapiran, "Intrusion detection system for wireless mesh network using multiple support vector machine classifiers with genetic-algorithm-based feature selection," *Comput. Secur.*, vol. 77, pp. 304–314, Aug. 2018.

[42] R. Abdulhammed, M. Faezipour, H. Musafer, and A. Abuzneid, "Efficient network intrusion detection using PCA-based dimensionality reduction of features," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Jun. 2019, pp. 1–6.

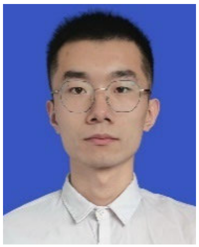
[43] L. Yang, A. Moubayed, I. Hamieh, and A. Shami, "Tree-based intelligent intrusion detection system in Internet of Vehicles," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[44] A. Rosay, F. Carlier, and P. Leroux, "Feed-forward neural network for network intrusion detection," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–6.

[45] S. Azodolmolky, J. Perelló, M. Angelou, F. Agraz, L. Velasco, S. Spadaro, Y. Pointurier, A. Francescon, C. V. Saradhi, P. Kokkinos, E. Varvarigos, S. A. Zahr, M. Gagnaire, M. Gunkel, D. Klonidis, and I. Tomkos, "Experimental demonstration of an impairment aware network planning and operation tool for transparent/translucent optical networks," *J. Lightw. Technol.*, vol. 29, no. 4, pp. 439–448, Feb. 2011.

SHAOQIANG WANG was born in Changchun, Jilin, China, in 1976. He received the master's degree in engineering from Jilin University, in 2007. He is currently a Professor with the School of Computer Science and Technology, Changchun University. He has hosted or participated in more than ten provincial and ministerial scientific research projects and published more than ten articles on research results. His main research interests include information security and data analysis.

YIZHE WANG was born in Yantai, Shandong, China, in 2000. He received the bachelor's degree in engineering from Qingdao University of Technology, in 2023. He is currently pursuing the degree with the School of Computer Science and Technology, Changchun University. His main research interests include the Internet of Vehicles and information security.



BAOSEN ZHENG was born in Qingdao, Shandong, China, in 1998. He received the bachelor's degree in engineering from Shandong Police College, in 2021. He is currently pursuing the degree with the School of Cyber Security, Changchun University. His main research interests include information security and data analytics.

JIAHUI CHENG was born in Nanyang, Henan, China, in 2001. He received the bachelor's degree in software engineering from Wenhua College, in 2023. He is currently pursuing the degree with the School of Cybersecurity, Changchun University. His main research interest includes information security.

YU SU was born in Lianyungang, Jiangsu, China, in 2000. He received the bachelor's degree in engineering from Hebei Institute of Environmental Engineering, in 2023. He is currently pursuing the degree with the School of Computer Science and Technology, Changchun University. His research interests include big data analysis and privacy protection.

YINFEI DAI is currently a Professor. She is a Master's Supervisor with the School of Computer Science and Technology, Changchun University. She presided more than Jilin Provincial Science and Technology Department of Key Scientific and Technological Research Projects Two, the Ministry of Education Chunhui Program Cooperation Project One Jilin Provincial Department of Education Research Projects Two, as a major participant in the provincial and ministerial-level research projects 16; published more than 40 articles, including the first author of the identity of the relevant areas of the public published 21 academic articles; involved in a number of enterprises and institutions with the cooperation of research and development projects; apply for and authorize the patent. She has applied and authorized 13 patents and 32 copyrights.