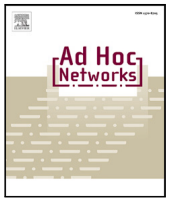
Ad Hoc Networks 153 (2024) 103330

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc



Intrusion detection system for cyberattacks in the Internet of Vehicles environment☆

Mohamed Selim Korium [a,*], Mohamed Saber [b], Alexander Beattie [a], Arun Narayanan [a], Subham Sahoo [c], Pedro H.J. Nardelli [a]

[a] Department of Electrical Engineering, School of Energy Systems, LUT University, Lappeenranta, Finland
[b] Department of Computer Science, Cairo University, Egypt
[c] Department of Energy, Aalborg University, Denmark

ARTICLE INFO

Dataset link: https://www.unb.ca/cic/

Keywords:
Internet of Vehicles
Intrusion detection system
Machine learning
Hyperparameter optimization
Learning curves
Overfitting

ABSTRACT

This paper presents a novel framework for intrusion detection specially designed for cyberattacks, such as Denial-of-Service, Distributed Denial-of-Service, Distributed Reflection Denial-of-Service, Brute Force, Botnets, and Sniffing, on vehicles that are situated in the Internet of Vehicles environment. We propose an intrusion detection system based on machine learning that is capable of detecting abnormal behavior by examining network traffic to find unusual data flows. In this paper, we have presented a strategy for intrusion detection through a careful evaluation and selection of the most effective techniques for the following steps of the machine learning process: (i) data preprocessing by using Z-score normalization that preserves the data distribution for the proposed method and handles outliers; (ii) feature selection by using a regression model that simplifies the model complexity and reduces the execution time; and (iii) model selection and training – Random Forest, Extreme Gradient Boosting, Categorical Boosting, Light Gradient Boosting Machine – with hyperparameter optimization to control the behavior in the training phase and to prevent overfitting. The effectiveness of the proposed solution is demonstrated by extensive numerical experiments carried out using the well-known standard datasets CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019, both separately and merged. We achieved a high accuracy above 99.8% within a running time of 46.9 s and 0.24 s detection time for the three combined intrusion detection system datasets, thereby showing that the proposed intrusion detection system outperforms the previous methods introduced in the literature.

1. Introduction

1.1. Background

Intelligent transportation systems (ITSs) have received significant research attention recently due to their potential to enable smart automated services in the transportation sector. By combining wireless devices with sensing technologies and advanced information and communication technologies (ICTs), ITSs can tackle many problems occurring during the transportation of goods and people, for example, safety, travel time, and harmful emissions [1]. ITSs are applicable to different means of transportation, such as airplanes, ships, trains, trucks, buses, and cars.

All deployments of ITSs rely on reliable communication technologies such as satellite and cellular networks [2]. In general, there are three main types of vehicular communication – vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-everything (V2X) – that enable road safety applications, such as forwarding collision warnings, lane change warning/blind spot warning, and emergency electric brake light warning through instant communication. V2X is a technology that facilitates communication between vehicles and other infrastructures. V2V is a type of two-way communication that allows vehicles to share data, such as velocity, current location, and destinations, with other vehicles. The messages delivered or received in V2V may also include information on nearby moving cars, helping the driver to immediately recognize cars in their blind spot. V2I, on the other hand, is a type of two-way communication that enables vehicles to connect and share data with external entities, such as traffic

☆ This paper is partly supported (i) by the Research Council of Finland (former Academy of Finland) via: (a) FIREMAN consortium CHIST-ERA/n.326270, (b) EnergyNet Fellowship n.321265/n.328869/n.352654, (c) mobility grant n.339541 and (d) X-SDEN project n.349965; and (ii) by Jane and Aatos Erkko Foundation via STREAM project.

* Correspondence to: P.O.Box 20. FI-53851 Lappeenranta, Finland
E-mail address: Mohamed.Korium@lut.fi (M.S. Korium).

https://doi.org/10.1016/j.adhoc.2023.103330
Received 19 June 2023; Received in revised form 26 September 2023; Accepted 14 October 2023
Available online 3 November 2023

signals, parking spaces, bicycles, and speed limits. V2I also includes radio communications that report the surrounding environment within a few kilometers of a vehicle's location [3]. In addition to these types, two new types of vehicular communications have recently emerged: Vehicle-to-Ecosystem (V2E) and Vehicle-to-Surroundings (V2S). V2E takes place between vehicles and external services such as satellite-based locations. This service can be a one-way service as in the case of the Global Positioning System (GPS) or a two-way service as requests for navigation services. V2S refers to a two-way communication type between vehicles and central control systems [4]. In general, all these different classes of vehicular communications can be considered to be part of a larger category, Internet of Vehicles (IoV).

Vehicles in the IoV environment, such as automated vehicles (AVs), are equipped with computational image sensors that give a 360-degree surround vision; actuators that control the vehicle motion to assist the driver; light detection and ranging (LiDAR); and radio detection and ranging (RADAR). These different sensors must perform under challenging conditions with ultrahigh reliability (approaching 100%) to enable the vehicles to sense and understand the surrounding road conditions. These diverse types of data have the potential to be used to improve road transportation safety, mobility flow, and environmental benefits by capturing and sharing data between vehicles and infrastructure. This would also enable different applications such as travel assistance, adaptive cruise control, and corporate fleet management [5]. In the near future, it is expected that the number of vehicles will increase, leading to an increase in vehicular communications and associated sensors. This will bring new vulnerabilities related to the cyber domain of the vehicles that are still lacking effective security mechanisms such as firewalls and gateways to avoid different types of cyberattacks. Cyberattacks have physical consequences, including serious threats to human lives when, e.g., the attacker takes command of the vehicle or when misguiding data are introduced to affect the decision process of the navigation algorithm. Attackers may use many different methods to control most or all the features of the vehicles, such as replacing a current video that was captured by the vehicle for image processing with a false stream.

To attack GPS signals, two kinds of actions can be identified—*jamming* (intentional interference) and *spoofing* (false identification) [6]. Jamming attacks are usually associated with the direct generation of radio signals to reduce the signal-to-noise ratio of the GPS signal. Spoofing attacks refer to the following cases: (i) inclusion of a falsely authorized GPS device; (ii) imitation of an image sensor used to steal data; and (iii) using an attacked GPS to provide the nodes with different or fake GPS information to disseminate malware or circumvent access control systems by obtaining security data. Both attacks are used to manipulate and send incorrect or fake GPS signals to a confused GPS device in vehicles. Attackers may synchronize (Syn) on the signals received from the satellite to make them even weaker. Following this, they can increase the frequency of the phishing signal that makes the vehicle's GPS lock on the fictitious signals, thereby dragging the vehicle to an incorrect position. LiDAR is also vulnerable to spoofing attacks that trick the system into thinking that there is an obstacle in the surrounding area that the vehicle must avoid. The attack is carried out by sending a signal to the vehicle's LiDAR system at the nanosecond level. This attack provides the attacker with the capability to interfere with the LiDAR structure or signal to coax it into not sensing any obstacle [7].

Inter-vehicle communications are also vulnerable to attacks with the most common ones being denial-of-service (DoS), distributed DoS (DDoS), and distributed reflection DoS (DrDoS) attacks. All these attacks are based on compromising and flooding the server with internet traffic or sending malicious requests to various components and communication networks in order to disrupt the normal functioning of vehicle systems such as the internal or external communication or navigation. Usually, a DoS attack is carried out by one or a small number of attackers with the intention of overloading the vehicle's systems and making it much slower. A DDoS attack is launched by multiple compromised devices that directly target the victim's resources leading to a severe service disruption and making it harder to trace; this is also known as the Botnets attack. A DrDoS attack is launched by sending requests to the vehicle communication devices that respond with larger volumes of data to a spoofed target. This attack depends on amplification techniques rather than the compromised devices as in the DDoS attack; therefore, it is more intricate to detect and more deceptive since the attacker appears legitimate until the goal is reached [8].

Network sniffing, brute force, infiltration, and web attacks share similarities in gaining unauthorized access but in different ways due to their different fundamental characteristics. Network sniffing works by monitoring and eavesdropping on the network communication that may hold sensitive data, such as the vehicle's location and the external communication servers. Brute force attacks focus on exploiting the communication and control mechanisms unique to connected vehicles and compromise the security of IoV-related services. Infiltration attacks involve gaining unauthorized access to vehicle systems by hacking into a vehicle's onboard computer, leading to the attacker taking direct control of the vehicle's components. Web attacks target the web services of the vehicles that the user may use as a web application. These attacks work by injecting malicious code into the infotainment system to gain control of the internal communication system in the vehicle or by compromising the user's data. These attacks demonstrate that more techniques are needed for robust sensor technologies to ensure adequate sensor data quality [9].

As vehicles become more integrated with the Internet and communication technologies to enable advanced features such as real-time traffic management, V2V, V2I, V2X, and VS2, they also become more vulnerable to external and internal cyberattacks. Therefore, building an Intrusion Detection System to identify such attacks plays a crucial role in ensuring the security and safety of connected vehicles with surrounding environments by detecting unauthorized agents when applying injection attacks to control a vehicle remotely or from accessing the vast amount of sensitive and personal data that is generated by the IoV, and by detecting any unauthorized access, especially during data transmission between the vehicles and the communication technologies [10].

1.2. Related research, its limitations, and our contribution

In the context of IoV, previous studies have focused simply on how a vehicle can ensure resilient operation even under an ongoing cyberattack only, as in [11]. Most prior works have reviewed security requirements, different possible security attacks, and countermeasures to overcome these attacks, without explicitly discussing how to *detect* them [9]. Further, the relatively few studies that have focused on the detection of cyberattacks either have tackled smaller problems or employed datasets that are now outdated. For example, the authors in [12,13] adopted k-nearest-neighbor (KNN) and artificial neural network models (ANN) only on an outdated dataset (NLS-KDD99) and obtained an accuracy level of 97%. In [14], machine learning (ML) algorithms were used to detect only a bot attack in a CIC-IDS-2017 dataset without considering other critical cyberattacks as DoS attacks.

Halbouni et al. [15] proposed a deep learning model, named CNN-LSTM, using convolutional neural network (CNN) and long short-term memory (LSTM) to construct a hybrid intrusion detection system model. The performance of the CNN-LSTM model was evaluated on the CIC-IDS-2017, UNSW-NB15, and WSN-DS datasets, and the accuracies of 99.64%, 94.53%, and 99.67%, respectively, were obtained. Although the CSE-CIC-IDS-2018 dataset contains fourteen different cyberattacks, the work by Halbouni et al. used a subset of the dataset including only four cyberattacks, namely a web attack, SSH-Patator, FTP-Patator, and Portscan, neglecting all types of DDoS cyberattacks. Abdulhammed et al. [16] used principal component analysis (PCA) to reduce feature dimensions from 81 to 10 features before adopting the random forest

algorithm. The performance of their model was evaluated on the CIC-IDS-2017 dataset, and the resulting accuracy was 99.6%.

Khan et al. [17] introduced a hybrid deep learning model called HCRNNIDS, which is a combination of CNN and recurrent neural network (RNN). Their aim was to use CNN to capture local features RNN to capture temporal features in order to improve the performance and prediction of an ID system. They claimed that the accuracy of their HCRNNIDS model was 97.75% on the CSE-CIC-IDS-2018 dataset. However, they had only used a subset of the dataset, and DDoS cyberattacks of four types – HOIC, LOIC-HTTP, SlowHTTPTest, and LOIC-UDP – were missing. Yang et al. suggested a decision tree (DT)-based IDS model [18], a multitiered hybrid (MTH-IDS) model [19], and an IDS framework called Leader Class and Confidence Decision Ensemble (LCCDE) [20] for autonomous and connected vehicles based on a controller area network (CAN) intrusion and the CIC-IDS-2017 dataset. According to the authors, their models achieved the highest accuracies of 99.86% [18], 99.88% [19], and 99.813% [20], compared with prior works, such as [9,11–17,21–28]. However, they did not show the learning curves of the models. It is crucial to include learning curves when the models achieve high accuracy in order to monitor the performance of the model over time, to generalize the behavior of the model, and to check and correct any tendency to overfit.

Seth et al. [29] proposed an IDS model using an ensemble classifier, a light gradient-boosting machine, and histogram-based gradient boosting algorithms (LightGBM+HBGB) on the CSE-CIC-IDS-2018 dataset. Their model achieved an accuracy of 96.97%. They claimed to have overcome the overfitting issue by using pruning and a random oversampling technique, even though it is shown in [30] that random oversampling produces replicas of the minority class samples, thereby increasing the possibilities of obtained over-fitted models. Javeed et al. [31] developed an SDN-enabled deep learning (DL)-driven model by using a hybrid technique comprising the Cuda-deep neural network LSTM (CuDNNLSTM) and Cuda-deep neural network gated recurrent unit (CuDNNGRU) algorithms for detecting a DDoS attack. According to the authors, the CuDNNLSTM + CuDNNGRU model achieved an accuracy of 99.74% in 9.33 ms; however, they have not indicated how they handled highly imbalanced data. As reported in [32], the CIC-DDoS-2019 dataset consists of highly imbalanced data, and therefore, the model may be overfitting the training samples from the underrepresented classes and may not generalize well to a new dataset [33]. In Assis et al. [34], the authors adopted many different models, such as gated recurrent units (GRU), LSTM, support vector machines (SVM), logistic regression (LR), KNN, and gradient descent (GD), on the CSE-CIC-IDS-2018 and CIC-DDoS-2019 datasets and they achieved accuracies of 97.1% and 99.6%, respectively. Their work only used a subset of the CSE-CIC-IDS-2018 dataset (DDoS and intrusion attacks), and the authors did not show how to handle highly imbalanced data.

In general, these prior studies on the intrusion detection problem suffer from the following limitations:

- Most contributions have used ML algorithms to detect bot attacks alone based on the CIC-IDS-2017 dataset, while others have used a subset of intrusion datasets to easily achieve a high accuracy without detecting all types of cyberattacks
- Many of these studies do not indicate how to handle highly imbalanced data often found in intrusion detection datasets
- The learning curves for monitoring the performance of the ML model over time and generalizing the behavior of the model have not been considered in most of the prior studies. Even though these existing models achieve a high accuracy, it is not clearly shown that overfitting is avoided
- Although Liang et al. [30] have proven that the synthetic minority oversampling technique (SMOTE) may increase the likelihood of the occurrence of overfitting, most prior works have used SMOTE and achieved high accuracies without considering learning curves
- Several studies have employed outdated datasets that contain old-fashioned, inflexible, unverified, and nonreproducible intrusions, such as NLS-KDD99, UNSW-NB15, and WSN-DS
- Most of the ML models yield high accuracies only on the specific dataset that was used to train their models. For example, models based on the CIC-IDS-2017 dataset perform poorly on the CSE-CIC-IDS-2018 dataset, even though it has about 38% similarity of attacks (this analysis is presented later)
- Only a few studies have considered the running, time, CPU time, and system time, although they are essential in cybersecurity, where time-sensitive activities must be carried out while detecting and thwarting IoV cyberattacks. Thus, it is critical to reduce the latency for detecting attacks on the IoV

To overcome the above issues, we propose an ML model that is tested on the following intrusion detection datasets: CIC-IDS-2017,[1] CSE-CIC-IDS-2018,[2] and CIC-DDoS-2019.[3] These datasets are more up-to-date external IoV cyberattacks, and they are openly available in two formats, namely packet capture (PCAP) and comma-separated values (CSV), maintained by the Canadian Institute for Cybersecurity using various attacks and tools. The datasets have about 80 features that are extracted by using the CICFlowMeter-V4 tool for CIC-DDoS-2019 and the CICFlowMeter-V3 tool for CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. In particular, we combined the three intrusion detection system datasets, thereby developing a comprehensive model that can be simultaneously applied to all three intrusion detection datasets. We demonstrated and compared the performance of our model solutions for these datasets. Further, the performance of our model is improved by leveraging a larger and more diverse set of training data as evidenced by our numerical results. Our model was able to successfully detect IoV cyberattacks with a high accuracy and a low execution time, as shown later in this paper. The main contributions of this paper are summarized as follows:

- We propose novel optimized ML-based algorithms that reduce the execution time for detecting various malicious IoV cyberattacks with a high accuracy
- We propose the use of a regression model for feature selection to assess the importance of such features and to remove noninformative or redundant predictors
- We discuss the uses of different oversampling techniques and determine the best suitable oversampling technique for each intrusion dataset
- We explore the impact of different types of hyperparameters to select the best model parameters for each dataset and discuss the ensemble learning techniques.
- We show, for the first time, how a high-accuracy model that overfits can cause a problem in detecting IoV cyberattacks by not generalizing properly, and propose a method to reduce overfitting while detecting IoV cyberattacks.
- We consider the execution time and learning curves to show that our model can perform and generalize well with different IoV cyberattacks and new datasets.
- We provide a comparison with recent works from the literature and demonstrate the effectiveness of the proposed framework that combines the three intrusion datasets, CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019. We show that our model is a good fit and outperforms the methods introduced in the literature in most cases

[1] https://www.unb.ca/cic/datasets/ids-2017.html
[2] https://www.unb.ca/cic/datasets/ids-2018.html
[3] https://www.unb.ca/cic/datasets/ddos-2019.html

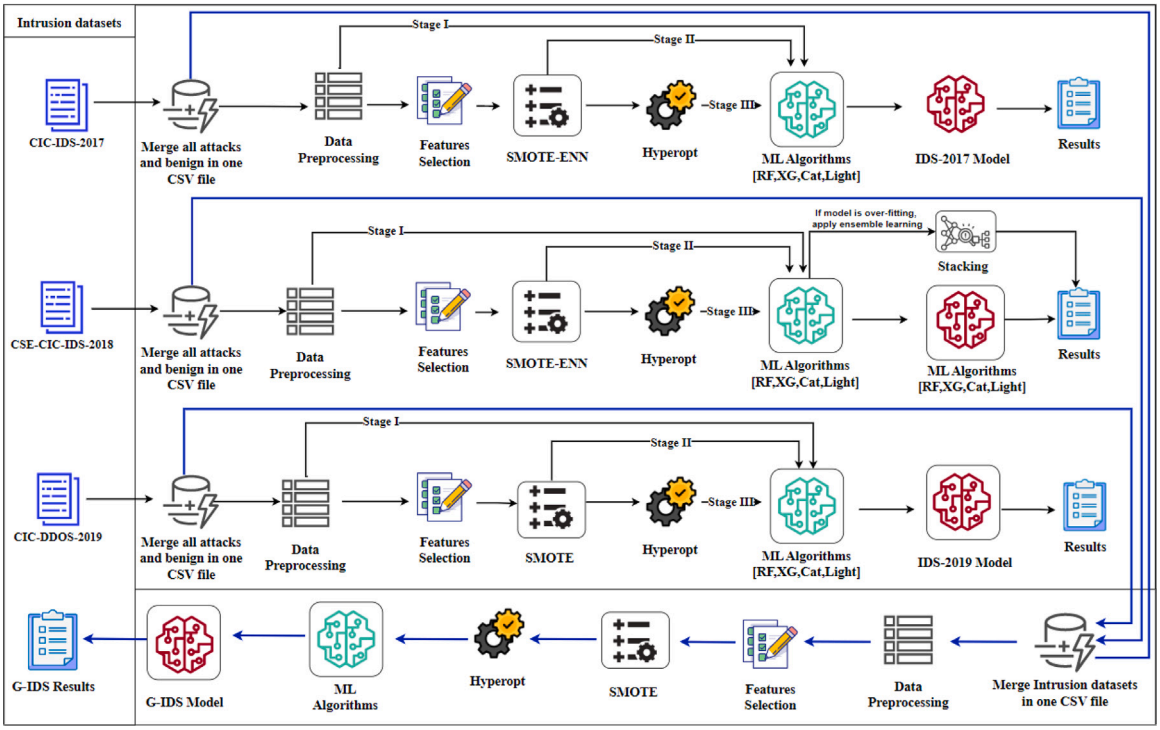


Fig. 1. Proposed IDS framework.

1.3. Paper structure

Section 2 explains the key concepts behind the implementation of the models, introduces the proposed framework, and comprehensive description of the characteristics of the datasets. Section 3 shows the model evaluation metrics at each stage with running time and the learning curves for the proposed models based on the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets. Section 4 introduces and discusses the model evaluation metrics and the learning curve for the merged intrusion dataset. Section 5 provides the rationale for considering the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets and presents numerical results, showing the advantages of our framework and the challenges encountered during the experimentation. Section 6 concludes the paper.

2. Proposed framework

2.1. System architecture and cyberattacks

The purpose of the present contribution is to develop a model-based method with highly accurate forecasting information of uncertainties related to cyberattacks in the IoV environment based on three different datasets: CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019. The proposed model architecture is depicted in Fig. 1 and the attacks considered in this work are shown in Table 1. The proposed IDS model's main stages consist of the following steps:

(1) Gathering external network traffic datasets in one comma-separated values file;
(2) Adopting a data preprocessing and feature selection approach;
(3) Applying different techniques to reduce problems caused by imbalanced data, such as SMOTE on the CIC-DDoS-2019 dataset and the merged intrusion data, and SMOTE based on the edited nearest neighbor algorithm (SMOTE-ENN) on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets;
(4) Applying hyperparameter optimization (Hyperopt) using Tree-structured Parzen Estimator algorithm to select the best model parameters;
(5) Using four different ML algorithms-Random Forest (RF), Extreme Gradient Boostoost (XGBoost), Category Boosting (CatBoost), and Light Gradient Boosting Machine (LightGBM) on the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets separately and merged;
(6) Implementing an ensemble learning technique as a potential solution, if the model is still overfitting even after tuning the hyperparameters; we show this approach later.

2.2. Intrusion datasets

The first stage in developing an IDS model is to gather both normal and abnormal states in network traffic data. To construct an efficient IDS, more data with more network properties should be collected, and the most common network characteristics and up-to-date datasets should be taken into account. Practical issues with the most often cited dataset, KDD-CUP99, include the following:

(1) The landscape of cyberattacks has changed considerably since 1999 (when KDD-CUP99 was published).
(2) Computer and network architectures are different with new technologies like Web3.
(3) Modern network protocols are not included in older datasets.
(4) Logging of network requests and data has changed significantly.

Therefore, we used intrusion datasets that contain both benign and the most up-to-date common IoV cyberattacks, such as DoS, DDoS, DrDoS, port-scan, and brute force, as shown in Table 1.

The data samples and files created by the network flow analysis in the intrusion datasets have a few limitations [35], which are listed below:

- There are some missing values and class labels. It is necessary to remove these outliers before conducting any further experiments
- The data samples created by the network flow analysis are saved in different files for each dataset. Processing these files is time-consuming because each file contains a significant number of data instances. The files can be merged to include each of the attack

Table 1
Traffic class for the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets.

CIC-IDS-2017 dataset		
Attacks	Traffic class	Record count
–	Benign	2 019 813
DoS	Hulk	231 073
	GoldenEye	10 293
	Slowloris	5 796
	SlowHTTPTest	5 499
	Heartbleed	11
	DDoS	128 027
Sniffing	PortScan	158 930
Brute Force	FTP-Patator	7 938
	SSH-Patator	5 897
Botnets	Bot	1 966
Infiltration	Infiltration	36
Web	Web Attack-Brute Force	1 507
	Web Attack-XSS	652
	Web Attack-SQL Injection	21
CSE-CIC-IDS-2018 dataset		
Attacks	Traffic Class	Record Count
–	Benign	13 484 708
DDoS	HOIC	686 012
	LOIC-HTTP	576 191
	SlowHTTPTest	139 890
	LOIC-UDP	1 730
DoS	Hulk	461 912
	GoldenEye	41 508
	Slowloris	10 990
Botnets	Bot	286 191
Brute Force	FTP-BruteForce	193 360
	SSH-Bruteforce	187 589
Infiltration	Infilteration	161 934
Web	Brute Force-Web	611
	Brute Force-XSS	230
	Web attack-SQL Injection	87
CIC-DDoS-2019 dataset		
Attacks	Traffic Class	Record Count
–	Benign	56 863
DrDoS	SQL-Server	4 522 492
	Net-BIOS	4 093 279
	User Datagram Protocol	3 134 645
	Network Time Protocol	60 132
	Light Directory Access Protocol	2 179 930
	Simple Service Discovery	2 610 611
	Trivial File Transfer Protocol	20 082 580
Intrusion merged dataset		
Attacks	Traffic Class	Record Count
	Benign	15 573 316
DoS	Slowloris	16 786
	Hulk	692 985
	GoldenEye	51 801
	Heartbleed	11
DDoS	LOIC-HTTP	576 191
	HOIC	686 012
DrDoS	NTP	1 202 642
	NetBIOS	4 093 279
	SSSDP	2 610 611
	UDP + lag	128 027
	TFTP	20 082 580
BruteForce	FTP-Patator	201 298
	SSH-Patator	193 486
Botnets	Bot	288 157
Sniffing	PortScan	158 930
	Synchronization	1 582 289

labels, but merging the cases of each attack type expands the dataset, which also takes longer to compute and process
- There are some attack types with invalid features, which include values like "Infinity" and "NaN" that are not suitable for ML algorithms
- The datasets are vulnerable to the problem of high-class imbalance, which can lead to a low accuracy and a high false positive rate

Therefore, the data have to be prepared for data preprocessing before it can be used for training and testing.

It is worth noting that this study was carried out on the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets both separately and merged. Our objective was to not only improve the classification accuracy but to also classify and detect IoV cyberattacks based on intrusion datasets using a single good fit model. This model will be evaluated using classification metrics and learning curves to ensure successful generalization to new IoV data. Note that only the attacks that are related to IoV cyberattacks are considered in this study.

2.3. Data preprocessing

The data preparation and cleaning process, which entails converting raw data into a format that ML algorithms can use to make detections, is a critical stage in ML [36]. Therefore, a few steps are required before conducting any further experiments to build an IDS with a high detection rate The first step is to remove unused columns, drop all incomplete and incorrect record data (missing values) because they affect the efficiency of the ML model, and remove the repeated features and columns. The second step is to convert the categorical and string values into numerical values and encode nonnumerical string values into integer values to be employed by the ML method. This conversion in the network traffic datasets is carried out by applying a $LabelEncoder$ from the sklearn library, which is an efficient tool for encoding the labels of the string and categorical features into numerical values [37]. By applying this conversion, the nonnumerical values are mapped into integer values between 0 and $n-1$, which makes them suitable for preprocessing by an ML algorithm. Even though the label tag is also a categorical element, it was not modified because original categories are required during the processing to distinguish the attack types in various forms and to test different approaches.

After the network traffic datasets are encoded by the label encoder, we need to normalize the data. Without normalization, even if the dataset has many advantageous features, one of those features might completely dominate the others. Both min–max normalization and Z-score normalization are used for preprocessing the data. The choice of normalization technique for IoV cyberattack detection depends on the data and the algorithm being used. We used min–max normalization for the CIC-IDS-2017 and the CSE-CIC-IDS-2018 datasets because the features had vastly different ranges. The min–max normalization technique lets all the features have the same scale by changing the values of the numeric columns ranging from 10,000 to 100,000 to numeric columns ranging from 0 to 1 without distorting the differences in the ranges of values or losing information. We implemented it by using a scaling formula called "min–max scaling" because ML training is often more efficient with normalized data [38]. Note that although the min–max method may eliminate some outliers, this does not affect the system performance because the detection task to be performed is designed to find long-term attacks [18]. In the CIC-DDoS-2019 dataset, the features have different ranges, but they are not as widely varied as for the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. Therefore, we used Z-score normalization for the CIC-DDoS-2019 and merged intrusion datasets [39].

Because of the high stream record for the CIC-DDoS-2019 and merged datasets, which may cause a high training time and low training efficiency, a few steps are required to reduce the size of the data and

the training complexity before conducting the data cleaning process Therefore, we used the *random sampling* technique to reduce the size of the data. The random sampling technique is a method of data collection designed to select a subset of data samples out of the original data. The selected data are chosen by random selection and equal probability [40]. The CIC-IDS-2019 and merged datasets are randomly sampled to 5% and 7%, respectively.

The crucial foundation for an ML model is to learn from the patterns by creating an algorithm so that the model can learn after the dataset has been preprocessed. Some datasets contain training and test data, and others are single unbundled datasets. The traffic datasets used in this study are single unbundled datasets. Therefore, after normalizing, we will prepare our model in a state that can be used by the ML by dividing our data into two sections, 80% training data and 20% testing data, by using Sklearn [41].

2.4. Feature selection

Feature selection strategies are used to minimize the complexity of the model by reducing the number of input variables and keeping those that are relevant and important for the task to both reduce the execution time and improve the performance of the model [42]. The performance of any model with a high-quality dataset depends on the feature selection; here, the required important independent features that have strong relationships with the dependent features are extracted [43]. Otherwise, the IDS model may have an accuracy of even less than 80%, which is unacceptable for an application designed to detect IoV cyberattacks. Without using feature selection, SMOTE or SMOTE-ENN to balance the data, the F1-score and the accuracy are low, as shown in Table 2 for the models based on CIC-IDS-2017 and CSE-CIC-IDS-2018. For the CIC-IDS-2019 dataset, the learning curve for the RF algorithm is overfitting, even though the accuracy and the F1-score reach 99.51%, as shown in Fig. 2.

Feature selection is not always applicable if the number of features in a dataset is too low; e.g, the NSL-KDD and UNSW-NB15 datasets have only 42 and 49 features, respectively [44]. On the other hand, each intrusion dataset has more than 80 features separately, which increases the model complexity and overfitting. In such a case, feature selection methods should be implemented for higher learning accuracy, lower execution time, and better model interpretability [45]. After data preprocessing, feature selection is carried out by an *RF regressor* (RFR) to select the important features for our goal. Feature selection by the RFR can be carried out by isolating each attack in a separate file or by creating one file that contains all types of attacks. Here, the entire datasets are used so that all the types of attacks are placed under one heading label in one merged file.

Random forest is an ensemble of many individual decision trees, and each function in this decision forest is given a weight of importance based on how useful it is in the construction of the decision tree [46]. If sufficiently informative features are not available for any type of attack, then the model will not be able to perform the ultimate task. If there are too many features, or if most of them are irrelevant, then the model will be trickier to train because it might go awry in the training process, which will impact the performance of the model. After the process is completed, these feature importance weights are compared and ordered [47].

The parameters in the RFR for feature selection include a *random state*, which is used to find the optimum split at each node, and *n-estimators*, which are the number of trees we want to build before taking the maximum voting or averages of predictions either to increase the detection power of the model or to make it easier to train the model. However, attackers mostly avoid using well-known ports; use created/fake IP addresses to avoid control; bypass operating system constraints; or avoid many other applications that are sent over the same port, thereby misleading the vehicle or the user [48]. Therefore, the feature selection is implemented by dropping columns with a low standard deviation, removing well-known ports, or using created/fake IP addresses and deceptive features, such as Flow ID, Source IP, Timestamp, Destination IP, and SimillarHTTP, to have more generic and invariant attributes to describe the attack [49]. As a result, the weights that are less than 0.015225, 0.01930, and 0.001598 are eliminated from the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets, respectively.

After eliminating the irrelevant features, the most relevant features shown in Table 3 serve as input for the models since they provide valuable insights into network traffic within the IoV context. They encompass a wide range of characteristics, including IP addresses used to identify the destination of the traffic user; port numbers used to

Table 2
Low-rate attack detection at stage I.

CIC-IDS-2017					
Attack	Class label	Model	Pre	Rec	F1
DoS	Hulk	XGBoost	1	0.13	0.22
		LightGBM	0.58	0.24	0.34
	GoldenEye	LightGBM	0.68	0.70	0.69
	Slowloris	LightGBM	0.79	0.49	0.61
	Slowhttptest	LightGBM	0.00	0.00	0.00
	Heartbleed	LightGBM	0.30	0.38	0.33
Sniffing	PortScan	XGBoost	0.99	0.64	0.78
		LightGBM	0.39	0.68	0.50
Brute-Force	FTP-Patator	XGBoost	0.99	0.68	0.81
		LightGBM	1.00	0.48	0.65
	SSH-Patator	XGBoost	1	0.50	0.67
		LightGBM	0.04	0.05	0.04
Botnets	Bot	RF	1.00	0.67	0.80
		XGBoost	1.00	0.25	0.40
		CatBoost	1	0.75	0.86
CSE-CIC-IDS-2018					
Attack	Class label	Model	Pre	Rec	F1
DDoS	HOIC	LightGBM	0.08	0.23	0.12
	LOIC-HTTP	LightGBM	0.75	0.42	0.54
	SlowHTTPTest	RF	0.76	0.52	0.62
		XGBoost	0.65	0.56	0.60
		CatBoost	0.76	0.52	0.62
	LOIC-UDP	LightGBM	0.00	0.00	0.00
DoS	Hulk	LightGBM	0.00	0.00	0.00
	GoldenEye	LightGBM	1.00	0.01	0.02
	Slowloris	XGBoost	1.00	0.62	0.77
Botnets	Bot	XGBoost	1.00	0.46	0.63
		LightGBM	0.29	0.45	0.35
Brute-Force	FTP	RF	0.72	0.88	0.79
		XGBoost	0.71	0.78	0.74
		CatBoost	0.72	0.88	0.79
Infilteration	Infilteration	RF	0.38	0.12	0.18
		XGBoost	0.00	0.00	0.00
		CatBoost	0.49	0.09	0.15
CIC-DDoS-2019					
Attack	Class Label	Model	Pre	Rec	F1
	Benign	LightGBM	0.99	0.50	0.66
DrDoS	UDP	RF	1.00	1.00	0.99
		XGBoost	1.00	1.00	0.99
		CatBoost	1.00	1.00	0.99
		LightGBM	0.99	0.74	0.84
	Syn	RF	1.00	1.00	0.99
		XGBoost	1.00	1.00	0.99
		CatBoost	1.00	1.00	1.00
		LightGBM	0.69	1.00	0.81
	LDAP	Random Forest	0.99	0.97	0.98
		XGBoost	1.00	0.98	0.99
		CatBoost	0.99	0.99	0.99
		LightGBM	0.00	0.00	0.00

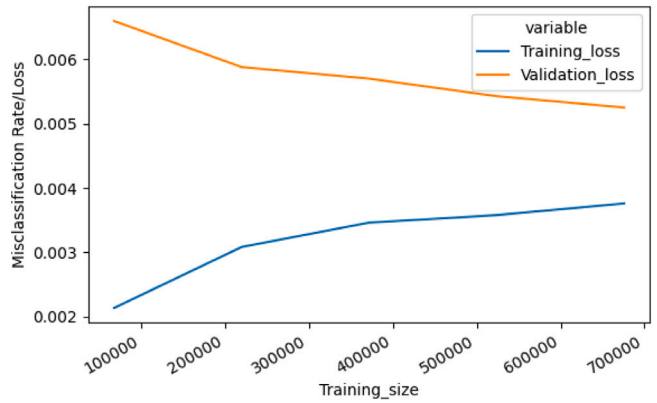


Fig. 2. Random forest algorithm based on CIC-DDoS-2019 is overfitting, and the running time is 129.6 s.

identify transport layer information; forward and backward packets that are used to measure the packet size distribution and variability in the forward and backward direction of a flow; and push, urgent, acknowledgment, synchronize, finish, and reset flags that are used to measure the urgency, acknowledgment, and connection frequency in the forward and backward directions of a transmission control protocol (TCP) and information. These characteristics help in understanding the patterns and behaviors of network traffic, which is crucial for detecting and mitigating intrusion attempts in IoV environments.

The impact of the number of features on the model performance depends on the characteristics of the dataset. Therefore, we employed learning curves in our studies to evaluate the performance of the models with different feature counts. This step was executed for the models that achieved exceptional accuracies exceeding 99% across different sets of features.

2.5. Synthetic minority oversampling technique with edited nearest neighbor

Imbalanced datasets are datasets with an unequal distribution of observations, meaning that one class label may have a significant number of observations while the others have fewer; this is common for classification problems. There are several methods to resolve this, e.g., oversampling or undersampling the majority class or their combination. After analyzing intrusion datasets, an imbalance of classes was detected because of its high frequency in the dataset. In CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets, the number of normal samples is higher than the percentage of attack samples, but this is the other way around for the CIC-DDoS-2019 and intrusion merged dataset (attack samples are higher than normal samples), causing a biased model and a low detection rate. In this study, an ENN algorithm, namely the SMOTE-ENN, was implemented on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets, and the SMOTE technique was implemented on the CIC-DDoS-2019 and the intrusion merged dataset to have an equivalent percentage between the classes. SMOTE and the hybrid technique SMOTE-ENN were chosen to create high-quality instances for minority classes and obtain cleaner synthetic samples to increase the model performance, rebalance the data frame, and increase the efficiency of the training algorithms [50].

Although SMOTE helps to overcome the overfitting problem, it could also lead to overfitting if the decision boundary between two samples gets tighter as in the case of the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets [30]. This can happen when minority class examples are interpolated; this can enlarge the minority class clusters and introduce artificial minority class examples too deeply into the space of the majority class. Therefore, we implemented the SMOTE-ENN on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. SMOTE-ENN works by increasing the number of samples in the minority class by using linear interpolation, whereas ENN reduces the number of samples in the majority class by removing noisy ones. Any sample with a class label that differs from at least two of its three closest neighbors is eliminated by SMOTE-ENN [51]. This process consumes a lot of time especially when the dataset has a massive number of samples, as shown in Table 1 for the CIC-DDoS-2019 and merged datasets. Therefore, SMOTE is implemented for the CIC-DDoS-2019 and merged datasets to reduce the execution time, and SMOTE-ENN is implemented for the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets.

In the data preprocessing stage, we used the random sampling technique to reduce the size of the CIC-DDoS-2019 and merged intrusion datasets. Remarkably, we observed that this sampling strategy reduced the degree of class imbalance, and as a result, we used the SMOTE algorithm because it is more effective when the overlap between the classes is low. For the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets,

Table 3
Weights of importance for both datasets.

CIC-IDS-2017		CSE-CIC-IDS-2018	
Features	Weight	Features	Weight
Source IP	0.110310	Init Fwd Win Byts	0.108490
Destination IP	0.058996	Fwd Seg Size Min	0.078871
Fwd Packets/s	0.055146	Flow Pkts	0.052779
Bwd Packets/s	0.050773	Fwd Pkts	0.045252
CWE Flag Count	0.046334	Fwd Header Len	0.042329
Bwd Packet Length Mean	0.045810	Totlen Fwd Pkts	0.032058
Fwd Packet Length Std	0.045281	Fwd Pkt Len Max	0.030269
Bwd Packet Length Min	0.040161	Flow Duration	0.030234
Destination Port	0.037626	Fwd IAT Tot	0.029618
Bwd Header Length	0.035552	Fwd IAT Mean	0.027873
Flow ID	0.025501	Flow IAT Min	0.026558
ACK Flag Count	0.023520	Subflow Fwd Byts	0.026248
Fwd Header Length	0.023285	Flow IAT Max	0.025326
Fwd Packet Length Min	0.020760	Flow IAT Mean	0.024267
Avg Bwd Segment Size	0.020015	Bwd Pkts	0.023788
Total Length of Bwd Packets	0.018726	Fwd IAT Max	0.023777
Flow Duration	0.017077	Init Bwd Win Byts	0.020853
Packet Length Std	0.015653	Fwd IAT Min	0.020844
Flow IAT Min	0.015628	Fwd Pkt Len Mean	0.020657
Fwd URG Flags	0.015225	Fwd Seg Size Avg	0.019307
CIC-DDoS-2019		**Intrusion merged dataset**	
Features	Weight	Features	Weight
Bwd URG Flags	0.093237	Flow Pkts/s	0.052546
Fwd Packets/s	0.077384	Bwd URG Flags	0.044819
Fwd URG Flags	0.068952	Flow Pkts/s	0.043001
Source Port	0.036341	Fwd Pkt Len Std	0.029613
Fwd Packet Length Min	0.031640	Fwd Pkt Len Min	0.024196
Min Packet Length	0.006561	Pkt Size Avg	0.022275
Bwd Avg Packets/Bulk	0.003123	Fwd Pkt Len Max	0.019412
Bwd PSH Flags	0.002786	Fwd Byts/b Avg	0.019357
Bwd Avg Bulk Rate	0.002546	ECE Flag Cnt	0.017763
Fwd Avg Bytes/Bulk	0.002128	Fwd Blk Rate Avg	0.013993
Fwd Avg Bulk Rate	0.002099	Bwd Blk Rate Avg	0.013023
Average Packet Size	0.002045	Bwd Byts/b Avg	0.011943
Flow Bytes/s	0.002042	Pkt Len Min	0.010669
Fwd Packet Length Std	0.001902	Fwd Seg Size Avg	0.010584
Avg Fwd Segment Size	0.001808	Fwd Pkts/b Avg	0.008577
Fwd Packet Length Max	0.001807	Init Fwd Win Byts	0.007957
Fwd Packet Length Mean	0.001759	Fwd Pkt Len Mean	0.007957
Bwd Avg Bytes/Bulk	0.001700	Bwd Pkts/b Avg	0.007926
Packet Length Mean	0.001598	PSH Flag Cnt	0.007781

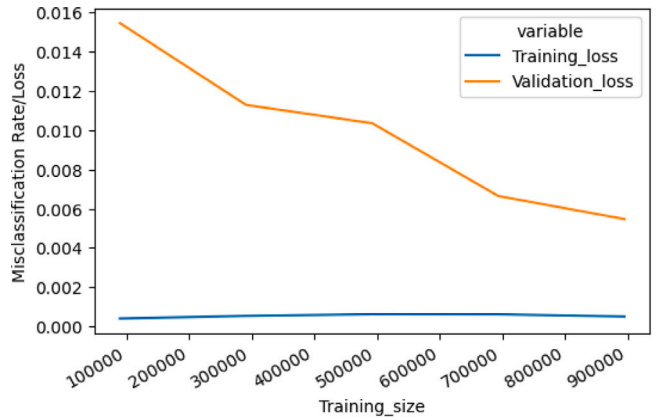


Fig. 3. Random forest model with 20 features based on the CIC-DDoS-2019 dataset is overfitting after feature selection and synthetic minority oversampling technique (SMOTE) are implemented.

we did not perform any sampling and observed that the degree of class imbalance was high. Therefore, we employed the SMOTE-ENN algorithm because it performs effectively when the overlap between the class imbalance is high [52,53].

As a result, we observed that the RF model based on the CIC-DDoS-2019 dataset is still overfitting after implementing the feature selection and SMOTE, as shown in Fig. 3.

2.6. Proposed machine learning approaches

Developing a model to identify diverse IoV cyberattacks can be considered a multiclassification challenge, i.e., a problem of classifying instances into one of three or more classes [54]. Unlike binary classification, it does not have any concept of normal and abnormal states. In this paper, this challenge is handled by implementing the following ML algorithms: RF, XGBoost, CatBoost, and LightGBM for the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets.

Random forest is a decision tree-based ML technique that is made up of many small decision trees, called estimators. This model then combines the classifier of the estimators to produce a more accurate prediction or detection, depending on the classification task [55]. XGBoost is an ensemble learning algorithm that is modified from the gradient-boosted decision trees (GBDT) algorithm. It uses the gradient boosting algorithm and the gradient descent algorithm to calculate the complexity of the leaf nodes of each tree and minimize the loss for finding the optimal score, respectively, to improve speed and performance [56]. A further development was introduced with the CatBoost model [57]: it is an implementation of ordered boosting, a permutation-driven alternative to the classic algorithm, which makes it an innovative algorithm for processing categorical features.

The above-described algorithms were created to counterbalance a detection shift caused by a special kind of target leakage present in all currently existing implementations of gradient boosting algorithms. Because of these features, the CatBoost algorithm reduces overfitting, improves model quality, and allows the entire dataset to be used for training. The most important benefit of CatBoost for our model is its stability when using default hyperparameters. With default hyperparameters, it produces stable results, which reduces the need for hyperparameter customization and tuning. However, these algorithms can be inconsistent with large, high-dimensional data that are present in the dataset. To solve this, the LightGBM model was proposed; it uses a Gradient Boosting Decision Tree, similar to XGBoost [58]. Two new techniques were proposed in the LightGBM model: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS uses only data instances with large gradients to determine the information gain to dramatically improve the algorithm speed while maintaining relatively accurate detection. The EFB technique enables feature reduction by bundling mutually exclusive features (i.e., they rarely take nonzero values simultaneously). In this study, the CatBoost and LightGBM ML algorithms were selected because they allow concurrent execution that cuts the model training time in half and improves efficiency during the model training phase. Moreover, the randomness in the creation phase of tree-based algorithms makes it possible to generate a robust ensemble model with more generalizability than other ML techniques. The above-described algorithms can be used for both predictive and detection purposes, depending on the classification task.

The four algorithms are found to perform effectively for Intrusion Detection Systems in the context of the IoV for several reasons [55–58]:

- These ML algorithms perform well with nonlinear and complex datasets since the IoV environments generate a large amount of data from internal and external communication technologies that might increase cyber threats. These algorithms perform effectively at detecting the intricate relationships within the data, making them effective at detecting IoV cyberattacks
- In the context of IoV, the vehicles continuously generate large amounts of data, which results in having an imbalanced dataset. These algorithms can handle imbalanced datasets and reduce the risk of false positives, especially Catboost and LightGBM which can handle real-time data streams and reliable intrusion detection
- Random Forest and boosting algorithms (XGBoost, LightGBM, and CatBoost) are known for their high detection accuracy and processing speeds, which are essential in cybersecurity.

2.7. Hyperparameter tuning

The parameters of the model are learned from the data, and hyperparameter tuning is used to find the best-fit for the learning algorithm [59]. It can be thought of as a model setting, which must be fine-tuned because the best values for one dataset will not be the same for other datasets to achieve a high precision and accuracy. These values are set before the learning process begins [60].

In this study, the search for the optimal hyperparameters for our IDS models is performed by random search cross validation (RSCV) because, as mentioned in [61], not only does the random search have all the advantages of grid search (conceptual simplicity, ease of implementation, and trivial parallelism), but it also samples only random points and does not give equal importance to every hyperparameter combination. In the intrusion detection system datasets, the network data have massive amounts of network traffic data, which is time-consuming, especially in the case of hyperparameter tuning. Therefore, a commonly used approach, the Bayesian optimization technique using the Tree-structured Parzen Estimator (TPE) algorithm in the Hyperopt library, is employed because of its sample efficiency [62]. Since there are many important hyperparameters in terms of accuracy, speed, and handling false negatives and false positives, the following hyperparameters were chosen in this study to prevent overfitting.

- For random forest, the quality of the split (Criterion) that includes “gini” or “entropy” for classification, the maximum depth of the tree (Max Depth), the maximum feature for the best split (Max Features), the minimum number of samples required for the leaf node (Min Sample Leaf), the number of trees in the forest (N-estimators), the minimum number of samples required to split (Min Samples Split), and the random state parameter are used to ensure that we have the same results each time we run the model
- For the gradient boosting algorithms (XGBoost, Catboost, Light): Column Subsampling by Tree to control the feature subsampling for building individual trees, learning rate controls the step size at each iteration, regularization terms applied to the weights of features (L1 Regularization Term on Weights), subsample to regulates a part of the training data to create tree, and column subsampling by tree parameter to regulate the chosen features to select a split candidate

2.8. Ensemble learning

In order to combine the algorithm models (RF, XGBoost, CatBoost, and LightGBM) with further accuracy improvement, an ensemble learning technique was implemented. Ensemble learning is a meta-approach to ML that combines outputs from different models by using three main classes: bagging, stacking, and boosting [63]. Bagging entails averaging the output models from many decision trees fitted to different samples of the same dataset. Stacking is the process of fitting multiple types of models to the same data and then using another model to learn how to integrate the output of multiple diverse detection models in the best possible way. Boosting includes sequentially adding ensemble members that correct prior model outputs of multiple diverse detection models and produce a weighted average [64].

The aim of using the stacking model in our study was to prevent overfitting [65] and to combine the output of the four models to have one model. Thus, we could evaluate the performance of the IDS-2018 model with different datasets and present the learning curves, as shown in Sections 4 and 5. It is worth noting that the ensemble learning techniques are not essential if the model shows no signs of overfitting. Therefore, we implement the ensemble learning technique, namely stacking, only if the model is overfitting after the hyperparameter optimization approach.

In this study, we implemented the following ML algorithms: RF, XGBoost, CatBoost, and LightGBM, and we then combined them by using a stacking model in which we ran the output of multiple models (base models) through a meta-learner that attempts to minimize the weakness and maximize the strengths of every individual model. This approach consists of two layers: the first layer contains trained base models, and its output is used as the input of the meta-learner in the second layer to create a powerful classifier. This classifier gives the highest accuracy among the four base models [63]. Random forest, CatBoost, and LightGBM were chosen in the proposed stacking method, and XGBoost was selected to be the meta-classifier of the stacking model for the CSE-CIC-IDS-2018 dataset

3. Performance evaluation for each stage

To evaluate the performance of the models for detecting external cyberattacks on IoV, we had to go through three stages (I, II, III). At each stage, we had to check the classification report and the learning curve, as shown in Fig. 4, so that the overfitting issue and IDS model performances are addressed in their current states. This is because, in some cases, data preprocessing is sufficient without implementing other approaches, such as feature selection and ensemble learning if the model is a good fit and achieves a high performance. Many unnecessary steps and approaches consume time, might reduce the efficiency of the model, and increase the chances of overfitting.

The performance measures applied to evaluate the proposed techniques are accuracy (ACC) that shows the accuracy of classification for each stage; detection rate (DR/recall) that is the ratio between the detected attack data and the total abnormal data; and harmonic precision–recall mean (F1-score) that is used as a statistical measure to rate the model performance since it depends on two factors, precision (Pre) and DR/recall. Further, the execution time of the models is calculated by using the "time" library that calculates the wall time, the user CPU time, the system time of the cell, and the total time (the sum of user time and system CPU time).

The test data can be employed to see whether our IDS model is able to recognize the patterns of each IoV external cyberattack. For evaluating our models, we focus on the F1-score metric as the accuracy of judgment. In most real-world classification situations, the data might have an imbalanced class distribution. Therefore, the F1-score is a preferable metric to evaluate our model. Further, the learning curve helps the user to track the improvement or deterioration of the learning performance of the model, and thus, it is critical to monitor the learning curve even if the model achieves a high accuracy. Learning curves are used to identify whether the learned model has overfitting or underfitting, or it is a good fit model [66].

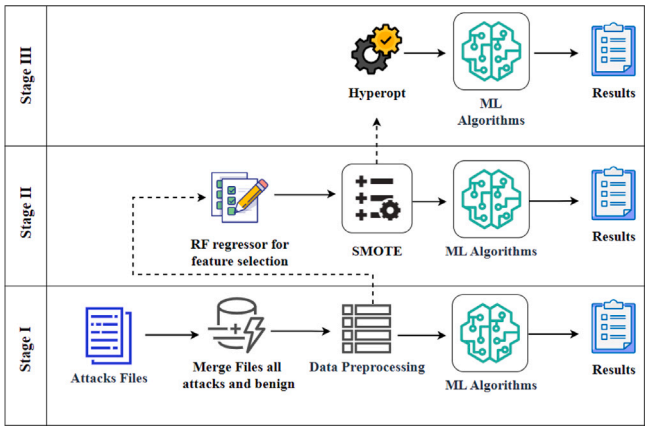


Fig. 4. Three stages to verify our model performance.

Table 4
Low detection rate for some IoV cyberattacks at stage II.

CIC-IDS-2017 at stage II					
Attack	Class label	Model	Pre	Rec	F1
Brute Force	FTP-Patator	XGBoost	0.98	0.89	0.94
	SSH-Patator		0.99	0.86	0.92
Dos	DoS GoldenEye		1.00	0.88	0.93
CSE-CIC-IDS-2018 at stage II					
Attack	Class label	Model	Pre	Rec	F1
Brute Force	FTP-Brute Force	RF	1.00	0.88	0.94
		XGBoost	1.00	0.88	0.94
		CatBoost	1.00	0.88	0.94
Infilteration	Infilteration	CatBoost	0.95	0.93	0.94
CIC-DDoS-2019 at stage II					
Attack	Class Label	Model	Pre	Rec	F1
DrDoS	LDAP	LightGBM	0.20	0.99	0.33

In stage I, we trained the model directly using the four algorithms – RF, XGBoost, CatBoost, and LightGBM – after data preprocessing. As shown in Table 2, the classification metrics for the models based on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets demonstrated that the models detected some attacks at a low rate. For the CIC-DDoS-2019 dataset, although most models (RF, XGBoost, and CatBoost) have a high F1-score after data preprocessing, the learning curve for the RF algorithm in Fig. 2 shows that the model is overfitting; in addition, the Light Directory Access Protocol (LDAP) attack has a low F1-score, as shown in Table 2.

In stage II, the verification performance was implemented after using RFR for feature selection and balancing the datasets using SMOTE technique for the CIC-DDoS-2019 dataset and SMOTE-ENN for the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets. The aim of stage II was to show if the chosen techniques (RFR, SMOTE, and SMOTE-ENN) are adequate or whether we need to tune the hyperparameters to improve the performance of the models and to reduce the overfitting. We observed that the models achieved a high accuracy for most IoV cyberattacks and a lower F1-score in some others, as shown in Table 4, and the RF model based on the CIC-DDoS-2019 dataset is still overfitting after implementing RFR and SMOTE algorithm, as shown in Figs. 3 and 5. An overall performance evaluation for stage II is shown in Table 5.

Based on these results, to achieve a better efficiency and a low execution time, we employed the TPE algorithm in Hyperopt at stage III to find the best-fit parameters that maximize the detection accuracy of

Table 5
Performance evaluation at stage II.

CIC-IDS-2017 at stage II					
Method	Pre	Rec	F1	ACC	Time (s)
RF	1.00	1.00	1.00	1.00	173.8
XGBoost	0.98	0.98	0.98	0.98	137.7
CatBoost	1.00	1.00	0.99	0.99	309.3
LightGBM	1.00	1.00	1.00	1.00	241
CSE-CIC-IDS-2018 at stage II					
Method	Pre	Rec	F1	ACC	Time (s)
RF	0.99	0.99	0.99	0.99	204.4
XGBoost	0.998	0.998	0.998	0.997	650.5
CatBoost	0.997	0.997	0.997	0.997	21.64
LightGBM	0.999	0.999	0.999	0.999	99.54
CIC-DDoS-2019 at stage II					
Method	Pre	Rec	F1	ACC	Time (s)
RF	0.9951	0.9951	0.9951	0.9951	194.4
XGBoost	0.9952	0.9952	0.9952	0.9952	966
CatBoost	0.9950	0.9950	0.9950	0.9950	360
LightGBM	0.984	0.987	0.984	0.986	142.2

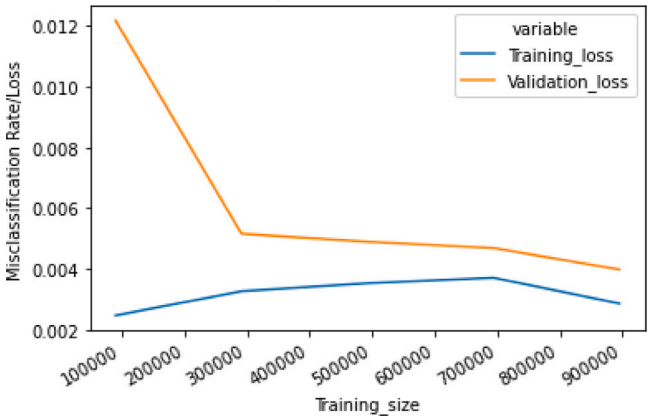


Fig. 5. Random forest with 75 features based on the CIC-DDoS-2019 dataset is overfitting after implementing the SMOTE.

Table 6
Performance evaluation at stage III.

CIC-IDS-2017 at stage III					
Method	Pre	Rec	F1	ACC	Time (s)
RF	1.00	1.00	1.00	1.00	89.60
XGBoost	1.00	1.00	1.00	1.00	684.86
CatBoost	1.00	1.00	1.00	1.00	9.51
LightGBM	1.00	1.00	1.00	1.00	82.02
CSE-CIC-IDS-2018 at stage III					
Method	Pre	Rec	F1	ACC	Time (s)
RF	0.9985	0.9984	0.9984	0.9984	4.72
XGBoost	0.9969	0.9968	0.9968	0.9968	9.152
CatBoost	0.9977	0.9976	0.9976	0.9976	16.34
LightGBM	0.9992	0.9992	0.9992	0.9992	32.22
CIC-DDoS-2019 at stage III					
Method	Pre	Rec	F1	ACC	Time (s)
RF	0.99475	0.99472	0.99473	0.99472	558
XGBoost	0.9944	0.9944	0.9944	0.9944	813.6
CatBoost	0.9931	0.9928	0.9929	0.9928	2,893.8
LightGBM	0.9933	0.9932	0.9933	0.9932	14.7

the model and reduce overfitting. The results indicate that the models have a high accuracy for IoV cyberattacks, as shown in Table 6, and the learning curves show that the overfitting issue has decreased, as illustrated in Figs. 6, 7, and 8.

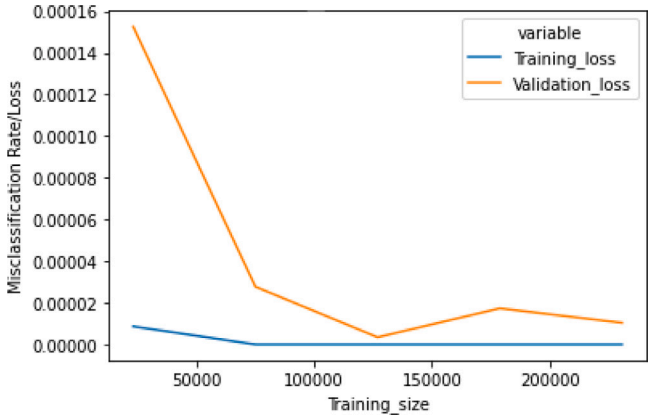


Fig. 6. LightGBM model based on the CIC-IDS-2017 dataset at a running time of 82.02 s.

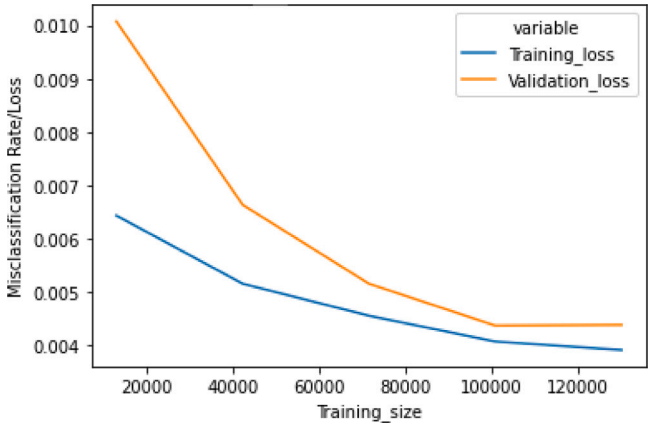


Fig. 7. XGBoost model based on the CSE-CIC-IDS-2018 dataset at a running time of 9.152 s.

It is noteworthy that in the CIC-IDS-2019 dataset, the models achieved a high F1-score that reached 99.51% for the RF model, 99.53% for the XGBoost model, 99.50% for the Catboost model, and 89.84% for the LightGBM model after the data preprocessing (without feature selection). Based on these results, it is worth considering the learning curves for the best-fit model with 20 and 75 features. The aim of this step is to determine the best-fit model either with 20 or 75 features. It was found that the RF model with 75 features and the Catboost model with 20 features are the best-fit models, and their learning curves demonstrate that the RF model, as shown in Fig. 8, provides a better fit than the CatBoost model with 20 features. In the CSE-CIC-IDS-2018 dataset, the RF and LightGBM models also achieved a high accuracy of 99.84% in 4.72 s and 99.92% in 32.22 s after implementing the TPE algorithm for hyperparameter tuning, but the learning curves for both models indicate that it is slightly overfitting. The model becomes overfitted when it memorizes the noise and fits the training set too closely. Therefore, it cannot generalize successfully to new data, and it will not be able to carry out the classification or detection tasks that it was designed for, as will be presented in Section 4.

4. Merged intrusion dataset

A high-quality IDS should not only achieve a high accuracy and low execution time, but it should also generalize well on new datasets. Further, it should be able to detect cyberattacks in different network environments of the same cyberattacks type. Based on the results presented in Section 3 for CSE-CIC-IDS-2018, we implemented ensemble

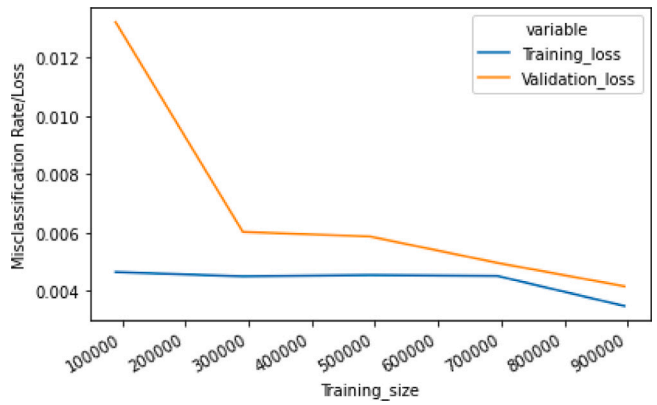


Fig. 8. Random forest model with 75 features based on CIC-DDoS-2019 and a running time of 558 s.

Table 7
Performance evaluation for ensemble learning.

IDS-2018 model					
Method	Pre	Rec	F1	ACC	Time (s)
Stack XGBoost	0.9993	0.9993	0.9993	0.9993	2.17

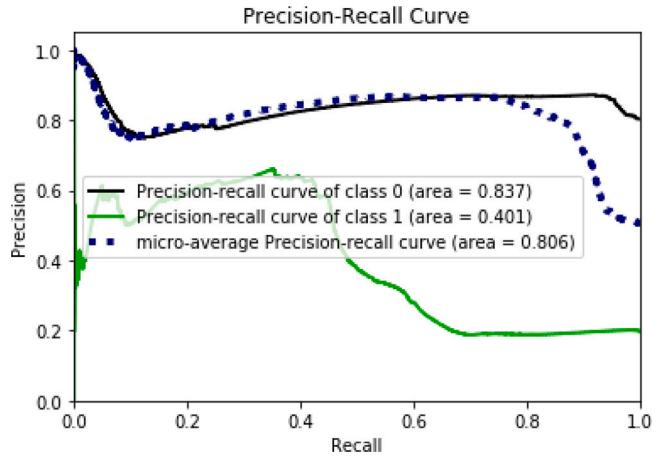


Fig. 9. Precision–Recall Curve IDS-2018 model on the CIC-IDS-2017 dataset.

learning (stacking the XGBoost model) to obtain one model, namely the IDS-2018 model, and then we test its performance on another dataset. The performance of the model after implementing ensemble learning achieved high accuracy and low running time, as shown in Table 7.

However, unfortunately, after the ensemble learning, the IDS-2018 model showed poor performance on the CIC-IDS-2017 dataset (a different network environment) showing a low Micro-averaged Precision–Recall, of 0.80, as shown in Fig. 9, even though the model evaluation metrics had showed a high accuracy for the IDS-2018 model on the CSE-CIC-IDS-2018 dataset.

These results suggest that:

(1) The IDS-2018 model used for the CSE-CIC-IDS-2018 dataset was not sufficient for the CIC-IDS-2017 dataset because the IDS-2018 model was slightly overfitting. Therefore, it will not generalize successfully to new data (such as CIC-IDS-2017), and it will not be able to carry out classification or detection tasks. Further, CSE-CIC-IDS-2018 is from a specific network environment, and therefore it is not easy to generalize the model well to different network environments.
(2) The CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets have different types of attacks, differing from each other by around 38%.

Table 8
Performance evaluation on the intrusion merged dataset after data preprocessing.

Intrusion merged dataset					
Attacks	Class label	F1-score			
		RF	XGB	Cat	LGB
	Benign	0.987	1.00	0.994	0.280
DoS	Slowloris	0.998	0.999	0.999	0.435
	Hulk	0.997	1.00	0.998	0.679
	GoldenEye	0.999	0.999	0.999	0.00
	Heartbleed	0.998	0.999	0.999	0.394
DDoS	LOIC-HTTP	0.996	1.00	0.999	0.00
	HOIC	0.999	0.999	0.999	0.00
DrDoS	NTP	0.999	1.00	0.999	0.499
	NetBIOS	0.998	0.999	0.998	0.626
	SSSDP	0.999	1.00	0.999	0.00
	UDP + Lag	0.996	1.00	0.999	0.610
	TFTP	1.00	1.00	1.00	0.00
BruteForce	FTP-Patator	0.995	0.999	0.9960	0.747
	SSH-Patator	0.996	0.999	0.996	0.411
Botnets	Bot	0.997	0.999	0.998	0.496
Sniffing	PortScan	0.996	0.999	0.997	0.00
	Synchronization	0.999	0.999	0.998	0.649
Accuracy		0.997	0.999	0.998	0.649
Running time (s)		60	294	3271.8	53.1

(3) Some of the classes in both datasets occur rarely, and the decision boundaries for these classes in both datasets are sometimes complicated and unspecific.

To overcome this issue, we combined data from different network environments to build combined IDS models that can detect IoV cyberattacks from the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets. This combination starts from merging all the attacks and benign data for each dataset in one CSV file, as shown in Fig. 10.

Training a model on a combination of intrusion datasets may cost a massive amount of time, especially during hyperparameter tuning, which requires training the model multiple times to achieve a high accuracy [19]. Therefore, data preprocessing is carried out as described in Section 2.3: the size of the merged dataset is reduced by using the random sampling technique; features with no variance, such as timestamps, are dropped out; and the label encoder and Z-score normalization are applied. Although the accuracy of RF and the F1-score reach 99.74%, as shown in Table 8, the model is overfitting owing to the imbalanced nature of the dataset and the absence of the feature selection method.

After data preprocessing, feature selection is carried out by RFR, where the number of trees in the forest is 250. The aim is to eliminate the irrelevant data and use only the relevant data, as described in Section 2.4. As a result, the weights of importance less than 0.002967 are eliminated.

The merged dataset is imbalanced data because of the high frequency between the classes. Therefore, the SMOTE algorithm is implemented to have an equivalent percentage between the classes, as described in Section 2.5 After balancing the dataset and performing feature selection, we employed the TPE algorithm in Hyperopt on all models (RF, XGBoost, CatBoost, LightGBM) to reduce overfitting as described in Section 2.7. The results indicate that the models are a good fit, as shown in Fig. 11; the performance evaluation after tuning the hyperparameter is shown in Table 10.

5. Numerical results

To build a high-performance model for identifying the majority of IoV cyberattacks, we considered CIC-IDS-2017, CSE-CIC-IDS-2018, CIC-DDoS-2019, and their combination, the intrusion merged dataset

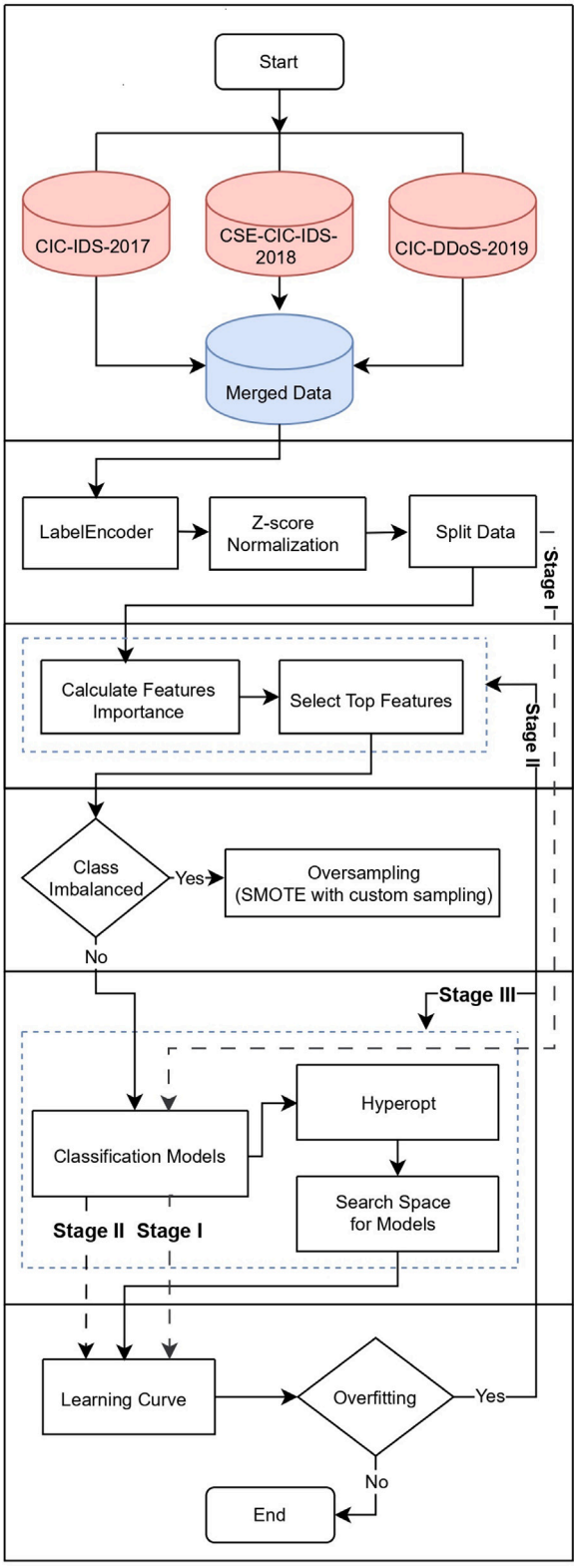


Fig. 10. Structure of the proposed models.

The results for stage I after implementing the data preprocessing as described in Section 2.3 show that more than 38% and 53% of the attacks were detected below the F1-score of 0.9 for the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets, respectively. The feature selection and the SMOTE and SMOTE-ENN algorithms are described in Sections 2.4 and 2.5, respectively. After the feature selection and the SMOTE-ENN algorithm were implemented, the results showed that the F1-score for the models based on the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets was improved by 37% and 51%, respectively, compared with stage I.

Fig. 11. Random forest model with 20 features based on the intrusion merged dataset; the running time is 46.9 s.

Table 9
Hyperparameter configuration for the random forest model.

Hyperparameter	Search range	Optimal value
Criterion	[gini , entropy]	entropy
Max depth	[5, 50]	36
Max features	[1, 29]	19
Min samples leaf	[1, 11]	3
Min samples split	[2, 11]	7
N-estimators	[10, 200]	27
Random state	[10, 200]	106

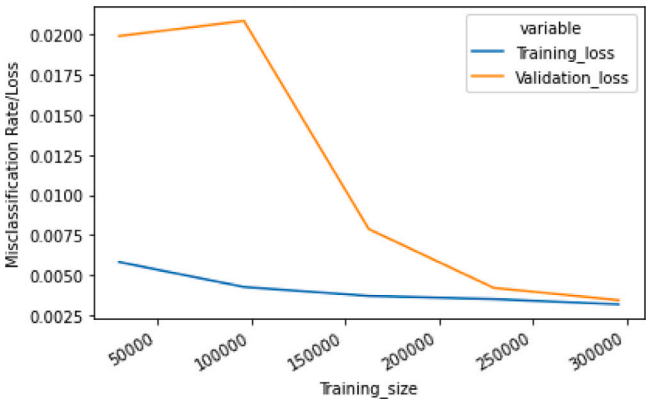


Fig. 12. CatBoost model with 75 features based on the intrusion merged dataset; the running time is 266.4 s.

After implementing the hyperparameter tuning as described in Section 2.7, the performance of most of the models improved by approximately 1%, as compared with stage II, and the models are a good fit, as shown in Figs. 6, 7, 8, and 11. The learning curve of a good fit model should indicate that the validation and training losses are close to each other. At the beginning of the learning curve for RF, the validation curve is gradually increasing instead of decreasing, as shown in Figs. 11 and 12. Therefore, to improve the validation curve, we considered the model with the smaller set of features, i.e., the RF model with 20 features shown in Fig. 11 (instead of the Catboost model in Fig. 12), and then, we added the regularization parameter 'Max Depth' into the hyperparameter tuning process using TPE algorithm in Hyperopt as shown in Table 9. As a result, the validation loss curve is decreasing

Table 10
Performance evaluation of the IDS models based on the intrusion merged dataset.

Intrusion merged dataset					
Attacks	Class label	F1-score			
		RF	XGB	Cat	LGB
	Benign	0.98785425	0.99418041	0.98882953	0.99207504
DoS	Slowloris	0.99895888	0.99895888	0.99843831	0.99895833
	Hulk	0.99791304	0.9986087	0.99773874	0.99912998
	GoldenEye	0.99975168	0.99975155	1.00	0.99950323
	Heartbleed	0.99946924	0.99956574	0.99908359	0.99975872
DDoS	LOIC-HTTP	0.99989204	0.99989202	1.00	1.00
	HOIC	0.99981665	1.00	0.99990832	0.99981658
DrDoS	NTP	0.99846272	0.99923136	0.99923136	0.99923136
	NetBIOS	0.99672814	0.99851102	0.99761905	0.99851013
	SSSDP	0.998125	0.99874922	0.99843701	0.99439601
	UDP+Lag	0.99846529	0.9988264	0.99837516	0.99918736
	TFTP	0.8894537	1.00	1.00	1.00
BruteForce	FTP-Patator	0.99601286	0.99601286	0.99601286	0.99588477
	SSH-Patator	0.9967151	0.99665467	0.99665256	0.99690442
Botnets	Bot	0.99824641	0.99824561	0.99831114	0.99834389
Sniffing	PortScan	0.99800857	0.99812926	0.99818972	0.99824995
	Synchronization	0.99916794	0.99923765	0.99854379	0.99923765
Accuracy		0.998008	0.998420	0.998019	0.998361
Precision		0.998012	0.998421	0.998020	0.998363
Recall		0.998008	0.998421	0.998019	0.998361
Average F1		0.998007	0.998420	0.998018	0.998361
Execution time					
Running time (s)		46.9	136.8	7.71	487.2
CPU time (s)		40.6	139	16	2411.4
System time (s)		0.222	0,281	1.92	1.52
Total time (s)		46.9	137	17.9	2412

Fig. 13. Random forest with 20 features and Hyperopt with a running time of 46.9 s.

instead of increasing at the beginning of the learning curve, as shown in Fig. 13 as compared with Fig. 11.

In our study, we creatively used and combined existing machine learning approaches to effectively detect IoV cyberattacks with a high accuracy and a low execution time, as shown in Table 11. Further, to ensure a fair and valid comparison with other methods, we compared our performance methods with different methods by using the same dataset and employed the same evaluation metrics as in other studies, including Accuracy, Precision, Recall, and F1-score. These experiments were carried out on the HP ZBook Power G7 Mobile Workstation with an I7-10750H CPU (2.6 GHz and 6 cores) and 32 GB of memory using Scikit-Learn, Imbalanced-Learn, and gradient boosting libraries in Python.

The comparative analysis between our proposed model and the methods employed by previous researchers demonstrates that our proposed model achieved outstanding performance with high metrics (accuracy, F1-score, recall, and precision) as shown in Table 11 and without overfitting issues as shown in Fig. 13.

The high accuracies obtained by the ML models (RF, XGBoost, CatBoost, and LightGBM) presented in Tables 6 and 10 as well as the main results given in Table 11 are in line with the results in similar works in the literature. The main reasons for the high accuracies can be attributed to the fact that there is a large difference between the attack and normal patterns in the datasets. Thus, to be able to easily distinguish them, we trained the model on large datasets, which improves the generalizability, and further, we used high-quality feature selection methods and Hyperopt. In addition, we tackled the overfitting issue in the following manner. We chose the best features and removed misleading features that cause overfitting. Further, we balanced data to improve the generalizability and employed a regularization technique and cross-validation on the training set. Moreover, we adopted algorithms that improve the detection accuracy and control overfitting, and we fitted the appropriate tree booster parameters, such as learning rate and maximum depth of a tree, which reduce the overfitting. It is worth mentioning that cyberattacks that are not related to IoV cyberattacks were eliminated from the CIC-DDoS-2019 and merged datasets. Moreover, the ensemble learning technique was only applied on the CSE-CIC-IDS-2018 dataset after the learning curves showed that the models were overfitting The aim was to have one model that could be used for testing with different datasets to determine whether the overfitted IDS-2018 model could generalize well with a new dataset.

The potential limitations and challenges that we faced are related to the datasets that contain a significant number of features, posing the challenge of determining a strategy to select the optimal feature or to reduce the dimensionality by selecting a subset of features, as mentioned in Section 3. Also, there are too many important hyperparameters in terms of accuracy and speed that make it challenging

Table 11
Comparison between the latest proposed IDS.

Best model	Year	Dataset	No. of attacks	Accuracy	Running time (s)
KNN [14] CART [14]	2020	CIC- IDS-2017	Bot attack	99.79% 99.97%	Not available
CNN-LSTM [15]	2019	CIC- IDS-2017	6	99.64%	Not available
RF+PCA [16]	2019	CIC- IDS-2017	14	99.6%	41.66
FS Stacking [18]	2021	CIC- IDS-2017	6	99.82%	2774.8
MTH-IDS [19]	2022	CIC- IDS-2017	6	99.88%	1563.4
LCCDE [20]	2021	CIC- IDS-2017	6	99.813%	169.9
HCRNNIDS [17]	2021	CSE-CIC-IDS-2018	7	97.75%	Not available
LightGBM+HBGB [29]	2021	CSE-CIC-IDS-2018	6	96.97%	6.13
GRU [34]	2021	CSE-CIC-IDS-2018 CIC-DDoS-2019	DDoS and intrusion attacks	99.6% 97.1%	Not available
CuDNNLSTM + CuDNNGRU [31]	2021	CIC-DDoS-2019	8	99.74%	9.33 ms
LightGBM+Hyperopt	**2022**	**CIC-IDS-2017**	14	**99.99%**	**82.02**
XGBoost+Hyperopt	**2022**	**CSE-CIC-IDS-2018**	14	**99.68%**	**9.15**
RF+Hyperopt	**2022**	**CIC-DDoS-2019**	7	**99.47%**	**558**
RF+Hyperopt	**2022**	**Intrusion merged dataset**	16	**99.80%**	**46.9**

to choose the specific parameters that have a high influence on the model's performance to achieve high accuracy with no overfitting issue and handle the false positives and false negatives.

In our analysis of the merged dataset for intrusion detection, several trends and patterns were observed that demonstrated the effectiveness of our approach:

- The proposed models have successfully been trained to identify complex patterns and features of different types of underrepresented attacks (least samples) such as SQL Injection attack apart from the normal network behavior (benign traffic) by using oversampling techniques such as SMOTE and SMOTE-ENN that reduce the bias, the overfitting issue, and minimize the false negatives (FN) and false positives
- The proposed model has been trained on fewer features and hyperparameters to achieve high performance. This indicates that the model has good efficiency with a low incidence of false positives and can save computational resources and time
- The learning curves showed that the training score and the validation score converged to a similar value as the number of training samples increased. This indicates that the proposed model has good generalization ability and could perform well on unseen data since it is not overfitting, as shown in Fig. 13.

In addition to the main contributions presented in Section 1, the objectives and novel contributions of the paper are not limited to only improving the classification accuracy, but also the following:

- We demonstrated that reaching a high accuracy with an overfitting model as shown in Table 7 cannot generalize well with a new dataset, as described in Section 4
- We considered the classification metrics and the learning curves at each stage to clarify if the methods used for the proposed models reduce the overfitting or not, as shown in Sections 4 and 5
- We explained how to reduce the overfitting and change the characteristics of the validation loss curve (see Figs. 11 and 13)
- We demonstrated four different model solutions that can be used to detect CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 cyberattacks simultaneously with a high accuracy.

6. Conclusion

Vehicles in the Internet of Vehicles environment are vulnerable to various types of cyberattacks, which may cause a serious threat to human lives. Therefore, we need an efficient IDS to guarantee safe operations against cyberattacks. In this paper, we developed IDS models to detect cyberattacks in well-known imbalanced datasets that are related to the Internet of Vehicles environment. The numerical results and the learning curves demonstrate that the selected methods enhance the performance of the model, leading to improved detection capabilities when tested on unseen datasets Remarkably, the RF model with Hyperopt achieved a high accuracy of 99.82% in 46.9 s on a combined dataset that was obtained by merging the CIC-IDS-2017, CSE-CIC-IDS-2018, and CIC-DDoS-2019 datasets. Our results with accuracy values consistently close to 100% are well aligned with results published previously in the literature, e.g., [14–20]. In the future, we plan to focus on different datasets and applications, assessing whether such high performance levels can be also achieved in different contexts. We will use different approaches such as deep reinforcement learning and transfer learning as they are expected to tackle the complexities and dynamics of IDSs.

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 dataset is available for everyone on https://www.unb.ca/cic/.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.adhoc.2023.103330.

References

[1] G. Dimitrakopoulos, P. Demestichas, Intelligent transportation systems, IEEE Veh. Technol. Mag. 5 (1) (2010) 77–84.
[2] Y. Lin, P. Wang, M. Ma, Intelligent transportation system (ITS): Concept, challenge and opportunity, in: 2017 Ieee 3rd International Conference on Big Data Security on Cloud (Bigdatasecurity), Ieee International Conference on High Performance and Smart Computing (Hpsc), and Ieee International Conference on Intelligent Data and Security, Ids, IEEE, 2017, pp. 167–172.
[3] F. Arena, G. Pau, An overview of vehicular communications, Future Internet 11 (2) (2019) 27.
[4] C.W. Axelrod, Integrating in-vehicle, vehicle-to-vehicle, and intelligent roadway systems, Complex Syst. Stud. (2018) 25.
[5] J. Ondruš, E. Kolla, P. Vertal', Ž. Šarić, How do autonomous cars work? Transp. Res. Procedia 44 (2020) 226–233.
[6] Z. El-Rewini, K. Sadatsharan, D.F. Selvaraj, S.J. Plathottam, P. Ranganathan, Cybersecurity challenges in vehicular communications, Veh. Commun. 23 (2020) 100214.
[7] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q.A. Chen, K. Fu, Z.M. Mao, Adversarial sensor attack on lidar-based perception in autonomous driving, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 2267–2281.

[8] R.R. Nuiaa, S. Manickam, A.H. Alsaeedi, Distributed reflection denial of service attack: A critical review, Int. J. Electr. Comput. Eng. 11 (6) (2021) 5327.

[9] K. Ren, Q. Wang, C. Wang, Z. Qin, X. Lin, The security of autonomous driving: Threats, defenses, and future directions, Proc. IEEE 108 (2) (2019) 357–372.

[10] I. Ivanov, C. Maple, T. Watson, S. Lee, Cyber Security Standards and Issues in V2X Communications for Internet of Vehicles, IET, 2018.

[11] A. Chowdhury, G. Karmakar, J. Kamruzzaman, A. Jolfaei, R. Das, Attacks on self-driving cars and their countermeasures: A survey, IEEE Access 8 (2020) 207308–207342.

[12] P. Dini, S. Saponara, Analysis, design, and comparison of machine-learning techniques for networking intrusion detection, Designs 5 (1) (2021) 9.

[13] T. Saranya, S. Sridevi, C. Deisy, T.D. Chung, M.A. Khan, Performance analysis of machine learning algorithms in intrusion detection system: A review, Procedia Comput. Sci. 171 (2020) 1251–1260.

[14] K. Aswal, D.C. Dobhal, H. Pathak, Comparative analysis of machine learning algorithms for identification of BOT attack on the internet of vehicles (IoV), in: 2020 International Conference on Inventive Computation Technologies, ICICT, IEEE, 2020, pp. 312–317.

[15] A. Halbouni, T.S. Gunawan, M.H. Habaebi, M. Halbouni, M. Kartiwi, R. Ahmad, CNN-LSTM: Hybrid deep neural network for network intrusion detection system, IEEE Access (2022).

[16] R. Abdulhammed, M. Faezipour, H. Musafer, A. Abuzneid, Efficient network intrusion detection using pca-based dimensionality reduction of features, in: 2019 International Symposium on Networks, Computers and Communications, ISNCC, IEEE, 2019, pp. 1–6.

[17] M. Khan, HCRNNIDS: Hybrid convolutional recurrent neural network-based network intrusion detection system, Processes 9 (5) (2021) 834.

[18] L. Yang, A. Moubayed, I. Hamieh, A. Shami, Tree-based intelligent intrusion detection system in Internet of Vehicles, in: 2019 IEEE Global Communications Conference, GLOBECOM, IEEE, 2019, pp. 1–6.

[19] L. Yang, A. Moubayed, A. Shami, MTH-IDS: A multi-tiered hybrid intrusion detection system for Internet of Vehicles, IEEE Internet Things J. (2021).

[20] L. Yang, A. Shami, G. Stevens, S. De Rusett, LCCDE: A decision-based ensemble framework for intrusion detection in the Internet of Vehicles, 2022, arXiv preprint arXiv:2208.03399.

[21] P. Verma, A. Dumka, R. Singh, A. Ashok, A. Gehlot, P.K. Malik, G.S. Gaba, M. Hedabou, A novel intrusion detection approach using machine learning ensemble for IoT environments, Appl. Sci. 11 (21) (2021) 10268.

[22] W. Elmasry, A. Akbulut, A.H. Zaim, Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic, Comput. Netw. 168 (2020) 107042.

[23] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.

[24] Y. Yao, L. Su, Z. Lu, B. Liu, Stdeepgraph: Spatial-temporal deep learning on communication graphs for long-term network attack detection, in: 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering, TrustCom/BigDataSE, IEEE, 2019, pp. 120–127.

[25] 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.

[26] M.A. Ferrag, L. Maglaras, DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids, IEEE Trans. Eng. Manage. 67 (4) (2019) 1285–1297.

[27] E. Min, J. Long, Q. Liu, J. Cui, Z. Cai, J. Ma, Su-ids: A semi-supervised and unsupervised framework for network intrusion detection, in: International Conference on Cloud Computing and Security, Springer, 2018, pp. 322–334.

[28] J. Lee, K. Park, GAN-based imbalanced data intrusion detection system, Pers. Ubiquitous Comput. 25 (1) (2021) 121–128.

[29] S. Seth, K.K. Chahal, G. Singh, A novel ensemble framework for an intelligent intrusion detection system, IEEE Access 9 (2021) 138451–138467.

[30] X. Liang, A. Jiang, T. Li, Y. Xue, G. Wang, LR-SMOTE—An improved unbalanced data set oversampling based on K-means and SVM, Knowl.-Based Syst. 196 (2020) 105845.

[31] D. Javeed, T. Gao, M.T. Khan, SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT, Electronics 10 (8) (2021) 918.

[32] D.-C. Can, H.-Q. Le, Q.-T. Ha, Detection of distributed denial of service attacks using automatic feature selection with enhancement for imbalance dataset, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2021, pp. 386–398.

[33] Z. Li, K. Kamnitsas, B. Glocker, Analyzing overfitting under class imbalance in neural networks for image segmentation, IEEE Trans. Med. Imaging 40 (3) (2020) 1065–1077.

[34] M.V. Assis, L.F. Carvalho, J. Lloret, M.L. Proença Jr., A GRU deep learning system against attacks in software defined networks, J. Netw. Comput. Appl. 177 (2021) 102942.

[35] A. Thakkar, R. Lohiya, A review of the advancement in intrusion detection datasets, Procedia Comput. Sci. 167 (2020) 636–645.

[36] H. Chen, J. Wang, D. Shi, A data preparation method for machine-learning-based power system cyber-attack detection, in: 2018 International Conference on Power System Technology, POWERCON, IEEE, 2018, pp. 3003–3009.

[37] E. Bisong, Introduction to scikit-learn, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Springer, 2019, pp. 215–229.

[38] K.M. Ali Alheeti, K. McDonald-Maier, Intelligent intrusion detection in external communication systems for autonomous vehicles, Syst. Sci. Control Eng. 6 (1) (2018) 48–56.

[39] M. Kuhn, K. Johnson, et al., Applied Predictive Modeling. Vol. 26, Springer, 2013.

[40] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, A. Markham, Learning semantic segmentation of large-scale point clouds with random sampling, IEEE Trans. Pattern Anal. Mach. Intell. (2021).

[41] A. Géron, Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, 2019.

[42] A. Jović, K. Brkić, N. Bogunović, A review of feature selection methods with applications, in: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, Ieee, 2015, pp. 1200–1205.

[43] J. Nothman, Scikit-learn/forest.py, Comput. Netw. (2019).

[44] S. Choudhary, N. Kesswani, Analysis of KDD-cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT, Procedia Comput. Sci. 167 (2020) 1561–1573.

[45] J. Miao, L. Niu, A survey on feature selection, Procedia Comput. Sci. 91 (2016) 919–926.

[46] S. Karasu, A. Altan, Recognition model for solar radiation time series based on random forest with feature selection approach, in: 2019 11th International Conference on Electrical and Electronics Engineering, ELECO, IEEE, 2019, pp. 8–11.

[47] U. Grömping, Variable importance in regression models, Wiley Interdisc. Rev.: Comput. Stat. 7 (2) (2015) 137–152.

[48] R. Alshammari, A.N. Zincir-Heywood, A flow based approach for ssh traffic detection, in: 2007 IEEE International Conference on Systems, Man and Cybernetics, IEEE, 2007, pp. 296–301.

[49] W. Xue, J. Zhang, Dealing with imbalanced dataset: A re-sampling method based on the improved SMOTE algorithm, Comm. Statist. Simulation Comput. 45 (4) (2016) 1160–1172.

[50] X. Zhang, J. Ran, J. Mi, An intrusion detection system based on convolutional neural network for imbalanced network traffic, in: 2019 IEEE 7th International Conference on Computer Science and Network Technology, ICCSNT, IEEE, 2019, pp. 456–460.

[51] J. Luengo, A. Fernández, S. García, F. Herrera, Addressing data complexity for imbalanced data sets: Analysis of SMOTE-based oversampling and evolutionary undersampling, Soft Comput. 15 (10) (2011) 1909–1936.

[52] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

[53] G.E. Batista, A.L. Bazzan, M.C. Monard, et al., Balancing training data for automated annotation of keywords: A case study, in: WOB, 2003, pp. 10–18.

[54] A. Moubayed, M. Injadat, A. Shami, H. Lutfiyya, Dns typo-squatting domain detection: A data analytics & machine learning based approach, in: 2018 IEEE Global Communications Conference, GLOBECOM, IEEE, 2018, pp. 1–7.

[55] P.A.A. Resende, A.C. Drummond, A survey of random forest based methods for intrusion detection systems, ACM Comput. Surv. 51 (3) (2018) 1–36.

[56] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[57] A.V. Dorogush, A. Gulin, G. Gusev, N. Kazeev, L.O. Prokhorenkova, A. Vorobev, Fighting biases with dynamic boosting, 2017, CoRR abs/1706.09516, arXiv:1706.09516.

[58] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, LightGBM: A highly efficient gradient boosting decision tree, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc., 2017, URL https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[59] H.J. Weerts, A.C. Mueller, J. Vanschoren, Importance of tuning hyperparameters of machine learning algorithms, 2020, arXiv preprint arXiv:2007.07588.

[60] J. Korstanje, The random forest, in: Advanced Forecasting with Python, Springer, 2021, pp. 179–191.

[61] J. Bergstra, Y. Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2) (2012).

[62] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, F. Hutter, Efficient and robust automated machine learning, in: Advances in Neural Information Processing Systems. Vol. 28, 2015.

[63] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisc. Rev.: Data Min. Knowl. Discov. 8 (4) (2018) e1249.

[64] I. Syarif, E. Zaluska, A. Prugel-Bennett, G. Wills, Application of bagging, boosting and stacking to intrusion detection, in: International Workshop on Machine Learning and Data Mining in Pattern Recognition, Springer, 2012, pp. 593–602.

[65] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, S. Gordon, A tree-based stacking ensemble technique with feature selection for network intrusion detection, Appl. Intell. (2022) 1–14.

[66] C. Perlich, Learning curves in machine learning, 2010.

Mohamed Selim Korium is currently pursuing a doctoral degree in Electrical Engineering with the School of Energy Systems at LUT University (Lappeenranta, Finland). He is also a researcher of the Cyber–Physical Systems Group in the LUT School of Energy Systems at the Laboratory of Control Engineering and Digital Systems at LUT University, where he has been actively working in deep reinforcement learning for AVs and mobile robots. He received the B.Sc. degree in mechatronics and robotics engineering from the Egyptian Russian University, Egypt, and Mechanical Engineering from LUT University.

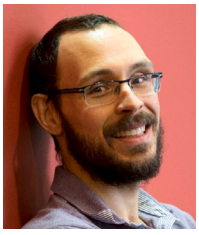
Mohamed Saber received the M.Sc. degree in Machine Learning and Data Analysis and Postgraduate Diploma in computer science from Cairo University, Egypt. His thesis work focused on network traffic analysis and cyberattack detection using machine learning algorithms. In the industry, he has more than seven years of experience in machine learning, software engineering, and systems analysis. His research interests are cyber–physical systems for smart grids, federated self-learning anomaly detection systems for IoT, deep reinforcement learning in complexes and computer networks, and AI-enabled blockchain smart contracts for a cyber-resilient energy infrastructure.

Alex Beattie graduated with distinction with a Fulbright-LUT University Master's degree in Mechatronic System Design. He graduated Summa Cum Laude from the University of Missouri with a B.Sc. in Computer Science. He has experience in working on interdisciplinary engineering projects in cyber–physical systems, electronics, and computer science domains. His thesis work focused on machine learning techniques for rare event and anomaly detection in streaming data. He is the founder of Gateway240 LLC, a rapid digital prototyping services company focused on sustainability in tech, enabling early stage entrepreneurial success, and building the future.

Arun Narayanan (M'14) received the B.E. degree in Electrical Engineering from the Visvesvaraya National Institute of Technology, Nagpur, India and M.Sc. in Energy Technology from Lappeenranta University of Technology (LUT), Finland, in 2002 and 2013, respectively. He subsequently completed his D.Sc. (Tech.) from the School of Energy Systems, LUT University, in 2019.

He is currently a postdoctoral researcher with LUT University, Lappeenranta, Finland, in the Cyber–Physical Systems Group. His research interests include renewable-energy-based smart microgrids, electricity distribution and markets, demand-side management, energy management systems, and information and communications technology. He focuses on applying optimization, computational concepts, and artificial intelligence techniques to renewable electrical energy problems.

Subham Sahoo (S'16-M'18) received the B.Tech. and Ph.D. degrees in Electrical and Electronics Engineering from VSSUT, Burla, India and Electrical Engineering at the Indian Institute of Technology, Delhi, New Delhi, India in 2014 and 2018, respectively. After completion of his Ph.D., he worked as a postdoctoral researcher in the Department of Electrical and Computer Engineering in the National University of Singapore during 2018–19 and in Aalborg University (AAU), Denmark during 2019–2020. He is currently an Assistant Professor at the Department of Energy, AAU, Denmark. He is a recipient of the Indian National Academy of Engineering (INAE) Innovative Students Project Award for his Ph.D. thesis across all the institutes in India for the year 2019. He was also a distinguished reviewer for IEEE Transactions on Smart Grid in the year 2020. His research interests are control, optimization, stability, and cybersecurity of power electronic dominated grids, and physics-informed machine learning tools for power electronic systems.



Pedro H.J. Nardelli received the B.S. and M.Sc. degrees in electrical engineering from the State University of Campinas, Brazil, in 2006 and 2008, respectively. In 2013, he received his doctoral degree from University of Oulu, Finland, and State University of Campinas following a dual degree agreement. He is currently Assistant Professor (tenure track) in IoT in Energy Systems at LUT University, Finland, and holds a position of Academy of Finland Research Fellow with a project called Building the Energy Internet as a large-scale IoT-based cyber–physical system that manages the energy inventory of distribution grids as discretized packets via machine-type communications (EnergyNet). He leads the Cyber–Physical Systems Group at LUT and is Project Coordinator of the CHIST-ERA European consortium Framework for the Identification of Rare Events via Machine Learning and IoT Networks (FIREMAN). He is also an adjunct professor at the University of Oulu in the topic of "communications strategies and information processing in energy systems". His research focuses on wireless communications particularly applied in industrial automation and energy systems. He received a best paper award of IEEE PES Innovative Smart Grid Technologies Latin America 2019 in the track "Big Data and Internet of Things". He is also IEEE Senior Member. More information: https://sites.google.com/view/nardelli/.