electronics

Article

CascadMLIDS: A Cascaded Machine Learning Framework for Intrusion Detection System in VANET

Argha Chandra Dhar [1], Arna Roy [1], M. A. H. Akhand [1,*] and Md Abdus Samad Kamal [2,*]

[1] Department of Computer Science and Engineering, Khulna University of Engineering and Technology, Khulna 9203, Bangladesh; dhar@cse.kuet.ac.bd (A.C.D.); roy1707018@stud.kuet.ac.bd (A.R.)
[2] Graduate School of Science and Technology, Gunma University, Kiryu 376-8515, Japan
* Correspondence: akhand@cse.kuet.ac.bd (M.A.H.A.); maskamal@gunma-u.ac.jp (M.A.S.K.)

Abstract: Vehicular ad hoc networks (VANETs) incorporating vehicles as an active and fast topology are gaining popularity as wireless communication means in intelligent transportation systems (ITSs). The cybersecurity issue in VANETs has drawn attention due to the potential security threats these networks face. An effective cybersecurity measure is essential as security threats impact the overall system, from business disruptions to data corruption, theft, exposure, and unauthorized network access. Intrusion detection systems (IDSs) are popular cybersecurity measures that detect intrusive behavior in a network. Recently, the machine learning (ML)-based IDS has emerged as a new research direction in VANET security. ML-based IDS studies have focused on improving accuracy as a typical classification task without focusing on malicious data. This study proposes a novel IDS for VANETs that offers more attention to classifying attack cases correctly with minimal features required by applying principal component analysis. The proposed Cascaded ML framework recognizes the difference between the attack and normal cases in the first step and classifies the attack data in the second step. The framework emphasizes that an attack should not be classified into the normal class. Finally, the proposed framework is implemented with an artificial neural network, the most popular ML model, and evaluated with the Car Hacking dataset. In addition, the study also investigates the efficiency of typical classification tasks and compares them with results of the proposed framework. Experimental results on the Car Hacking dataset have revealed the proposed method to be an effective IDS and that it outperformed the existing state-of-the-art ML models.

Keywords: cascaded model; intrusion detection; malicious data; neural network

Citation: Dhar, A.C.; Roy, A.; Akhand, M.A.H.; Kamal, M.A.S. CascadMLIDS: A Cascaded Machine Learning Framework for Intrusion Detection System in VANET. *Electronics* **2023**, *12*, 3779. https://doi.org/10.3390/electronics12183779

Academic Editor: Juan-Carlos Cano

Received: 17 July 2023
Revised: 21 August 2023
Accepted: 4 September 2023
Published: 7 September 2023

1. Introduction

A vehicular ad hoc network (VANET), or simply vehicular network, is a type of mobile ad hoc network that is specifically designed for communication between vehicles. It utilizes wireless communication innovations such as Wi-Fi, cellular, and dedicated short-range communication to enable communication between vehicles and the surrounding environment [1]. In VANETs, vehicles act as both senders and receivers of information and can dynamically create a network [2] without the need for a fixed infrastructure. This makes VANETs ideal for providing real-time information in situations where traditional communication networks may be unavailable, such as during natural disasters, road construction, and other types of network disruptions [3]. VANETs can be used to support a variety of applications, including road safety, traffic management, navigation and location-based services, entertainment, etc. Overall, VANETs represent an important technology for enhancing road safety and improving the driving experience by providing real-time information and communication capabilities.

The scope for the VANET is increasing day by day due to its self-aware system and being a wireless network [4]. High dynamicity, a sensitive information-sharing system, and time susceptibility make this network susceptible to different types of cyberattacks.

A cyberattack is an unwanted attempt to steal, hide, expose, manipulate, and alter information through illegally accessing a computer network or the internet. It can lead to cyberterrorism, war, or threats [5]. The availability, low cost, and rapid improvement of the internet play a significant role in our lives. But this rapid development is also expanding the scope of cyberattacks due to the expanding trends of remote work, availability of sensitive data, and insufficient and inefficient intrusion detection systems. Various cyberattacks are possible, including, but not limited to, denial of service, man-in-the-middle attacks, phishing attacks, cross-site scripting attacks, malware, birthday attacks, and intruder attacks [6]. As a remedy, different methods are being introduced to stop and detect cyberattacks which have been key research subjects in cybersecurity. Cybersecurity has been a crucial research area over the last decade since different levels of high-tech attacks are being employed by attackers [7]. Cybersecurity has received significant attention globally, with the ever-continuing expansion of internet usage and due to growing trends and adverse impacts of cybercrimes. High-profile cybercrimes have revealed the ease of spreading international cyberattacks, including disrupting businesses, corrupting sensitive data, stealing information, illegitimate access, etc. [8].

As a preventive measure against such attacks, different kinds of firewalls, antivirus systems, phishing detection [9], anomaly-based detection systems [10], and intrusion detection systems (IDSs) [11] have been introduced throughout the last decade. The enormous evolution of technology has raised the need for proper IDS products, and research evolving IDSs is increasing day by day for all networks. An IDS is a type of security system that aims to detect unauthorized access or malicious activities on a computer network. It monitors network traffic and system events to identify suspicious patterns or behavior that may indicate an attempted intrusion [12]. There are two main types of IDS: Network-based IDS (NIDS) and host-based IDS (HIDS) [13]. The NIDS is concerned with the entire network system, and it examines the activities and traffic of all the systems in the network. On the other hand, the HIDS is related to just a single system. As the name suggests, it is only concerned with the threats related to the host system/computer. IDSs use various techniques to detect intrusions, including signature-based detection, anomaly-based detection, and stateful protocol analysis. Signature-based detection uses patterns or autographs that have been predefined to recognize known attacks, while anomaly-based detection identifies intrusions by detecting deviations from normal behavior. Stateful protocol analysis is used to validate the state of a connection and detect malicious activities that violate the state.

Recently, due to their drastic growth and superior performance, machine learning (ML)-based autonomous systems have become the state of the art in cybersecurity [14]. ML is a data-driven approach, and appropriate data, i.e., a large amount of data and sufficient sample categories, are essential to develop a sensible ML model. The Car Hacking dataset [15] is one of the most popular IDS benchmark datasets. Diverse IDS models have been developed for VANETs using several ML methods, including deep convolution neural networks (CNNs) [15], generative adversarial net [16], deep neural networks (NNs) [17], ensemble NNs [18], and so on. The IDS studies for VANETs are focused on achieving better accuracy with a standardized dataset in general, overlooking solving the discrepancies of the datasets, time complexity, and efficiency of the methods. Only an accuracy measure is insufficient for VANETs as millions of packets go through the network daily. For this reason, if accuracy drops only 0.1%, it may cause a severe issue for the network.

Developing a more effective and efficient IDS for VANETs is crucial due to these networks' unique challenges and vulnerabilities. VANETs facilitate communication among vehicles and infrastructure, enabling advancements in road safety and traffic management. However, their dynamic and decentralized nature makes them susceptible to various security threats, including unauthorized access, data tampering, and malicious attacks. Current IDSs for VANETs often struggle to address these challenges adequately. They may suffer from high false positive rates, leading to unnecessary disruptions in traffic flow, or

they might miss sophisticated and rapidly evolving attack patterns. So, enhancing IDSs for VANETs is essential to ensure the security and reliability of vehicular communication.

A critical deficiency in recent studies on IDSs lies in their tendency to prioritize overall accuracy improvement without adequately addressing the challenge of differentiating attack data from normal patterns. While enhancing accuracy is undoubtedly crucial, it is equally important to ensure that attack data are not misclassified as normal behavior. This concern is especially pertinent in the context of VANETs, where the consequences of misclassifying malicious activities as normal can be catastrophic. Failing to distinguish attacks from legitimate communication undermines the fundamental purpose of IDSs, leaving the network vulnerable to various security threats. Therefore, future research efforts should not only focus on boosting accuracy but should also place significant emphasis on refining the ability of IDSs to accurately identify and respond to emerging attack patterns, safeguarding the integrity and safety of vehicular communication within VANETs.

A prevailing shortcoming in ML-based IDSs is their limited consideration of the necessity of every parameter in the classification process. These systems often utilize a multitude of parameters to achieve high accuracy, yet they may neglect to evaluate whether each parameter truly contributes to the accuracy or if some could introduce noise or redundancy. This oversight can lead to increased computational demands and reduced efficiency, particularly in resource-constrained environments. A more strategic approach would involve assessing the relevance of each parameter, optimizing feature selection, and ultimately crafting leaner and more efficient IDS models that preserve accuracy while minimizing unnecessary computational overhead.

Another important point is that most of the recent studies have prioritized complex models, as intricate models may excel across diverse scenarios, but it is essential to recognize that simpler models within a different framework can excel in efficiency and time-critical scenarios. Emphasizing context-specific solutions over complexity allows for effective intrusion detection while accommodating the inherent constraints of VANETs.

This study aims to develop a novel IDS that gives more attention to classifying attack cases correctly and categorizing attacks through a rigorous experimental process. A two-step process with a cascaded framework is proposed, where an ML model recognizes the attacks and another ML model classifies them into different sorts of attacks in consecutive operations. The framework emphasizes that an attack class should not go into the normal class by developing a simple, efficient ML model. Therefore, an appropriate cascading technique is introduced for adequately preparing the training set to learn attack attributes using samples of the original dataset. The contributions of the study are summarized as follows:

1. A Cascaded ML framework model is introduced to give more considerable attention to classifying attacks into the major classes in two steps;
2. While developing the framework, it is considered that the proposed model should be simple, but it should not compromise the accuracy, effectiveness, and time complexity;
3. The proposed framework applies the principal component analysis algorithm to remove irrelevant features and ensure fast and efficient detection.

The structure of the rest of the paper is as follows. Section 2 describes background studies and the literature review. Section 3 demonstrates the proposed cascaded neural network based on IDSs for VANETs. Section 4 presents experimental outcomes and performance comparisons with previous studies. Finally, Section 5 concludes the paper with a few remarks.

2. Background Studies and Literature Review

Vehicular networks are nowadays considered an innovative class of wireless networks. Different methods, such as ML algorithms, game theoretic approach, etc., have evolved to solve various issues of these networks. Recently, with radical growth and higher performance, ML-based autonomous systems have appeared as state-of-the-art approaches in the cybersecurity of vehicular networks.

The VANET was first introduced in 2001 as a "car to car" [14] communication network where communication and a network can be formed among cars so that each car knows information about other surrounding cars [19]. It has transferable vehicles which function as nodes to create a portable network. The VANET transfers each vehicle as wireless nodes, allowing them to connect 100–300 m apart [20]. If a car drops out due to signal range and network issues, another car can join the network; hence, a mobile internet can be created over a range. The VANET provides a range of opportunities for urban monitoring and data distribution aspects, allowing the VANET to work in a broader range and collect a large amount of data [21]. The VANET consists of many highly mobile vehicles, leading to rapid changes in the network topology. The VANET generally has a variety of vehicles with different communication capabilities, leading to a heterogeneous network. From an ITS [22] viewpoint, an intelligent vehicle must deal with road safety and traffic efficiency, including forward and blind-spot collision warnings, lane distribution warnings, rollover warnings for substantial vehicles, pedestrian detection, driver monitoring, etc. These self-organizing characteristics are often seen in VANETs, and therefore research on VANETs leans on the development of ITSs. Additionally, critical latency requirements and no power constraints make VANETs compatible with other wireless networks.

There are three main components in a VANET [23]: 1. Onboard unit (OBU), which is housed inside the vehicle and provides the interface for exchanging information with the roadside unit or other OBUs. It consists of a processor, read/write memory, user interface, and network device to communicate in a short-range area. 2. RSU, a static device fixed along the side of the road or at a junction, which can connect to the internet and deliver information to the users as soon as possible. 3. Application unit (AU), which is a dedicated device of the OBU to utilize the requests provided by the supplier to confirm safety.

The VANET is a developing paradigm, bringing enormous profits to society. The VANET mainly works in the environment of wireless networks, so sharing information can be crucial, as bogus information can cause serious accidents. Therefore, designing an efficient and trustworthy algorithm to secure VANETs is compulsory. Different types of attacks based on timing, network, types of vehicles, routing protocols, and monitoring are seen in VANETs [24–26]. Bogus information, timing attacks, counterfeit messages, masquerading, node impersonation, GPS spoofing, malware, unauthorized access, Sybil attacks, denial of service (DoS), black hole attacks, man-in-the-middle attacks, eavesdropping, etc. are some examples of prominent VANET attacks.

Several works have been reported on VANETs over the last few years. Typical machine learning algorithms multilayer perceptron (MLP) and decision tree (DT) were used to distinguish DoS and fuzzy attacks [27]. In [28], an unsupervised Kohonen self-organization map, a NN, was utilized to develop an anomaly-based detection system. The authors used the Car Hacking dataset and integrated the k-means algorithm to enhance the effectiveness of their model. The study in [29] used C4.5, a decision tree algorithm, for IDS development. In [30], a VANET security model is proposed that works in two stages: The trust estimation model, which adapts ML algorithms, and the decision model. In [31], an IDS is proposed based on a multivariate statistical approach and k-nearest neighbor (KNN) algorithm, which can detect attacks and alleviate their effect where possible. An ML-based IDS was developed in [32] to detect malicious attacks in a timely manner. The proposed ML model consists of classic random forest algorithms and clustering-based algorithms.

Recent studies on the security of VANETs have considered deep learning (DL) methods. A deep convolution neural network (CNN)-based IDS was suggested by [15] to protect the vehicles of a control area network (CAN). The authors experimented on real-time vehicles, which gave a significantly better result. This deep CNN model is the reduced form of a ResNet model. In [16], a generative adversarial net-based IDS was mentioned, which showed high performance in detecting unknown attacks. A DNN was used to build an IDS [17] to detect malicious packets through a probability distribution. The study in [18] considered an ensemble of neural networks (NNs) to classify different types of attacks in the dataset. The base classifiers of the ensemble are an autoencoder, a deep belief NN, a

deep NN (DNN), and an extreme learning machine. In particular, the detection and false alarm rates of the implemented ensemble were evaluated. The automated data mining tool WEKA was used to conduct experiments using J48, SVM, and naïve Bayes algorithms for classification.

The authors of [33] used two DL methods: Long short-term memory (LSTM) and a gated recurrent unit (GRU) as a hybrid network model for an IDS. A second dense layer connects these two methods to work as a whole model. They investigated their model using DoS attacks and the Car Hacking dataset. Conventional LSTM was considered in [34] to develop an IDS for VANETs. The authors generated a real-time dataset to train their model and a public dataset to evaluate the performance of the model.

LSTM and autoencoders were considered in [35] to develop an IDS for VANETs. The authors worked particularly on V2V and V2I structures to identify suspicious attacks. The LSTM layer captures different features in their model, and the autoencoder tries to regenerate traffic data. The authors employed a conventional CNN [36] to detect intrusive behavior in the VANET.

A feature extraction algorithm and a classifier based on an improved growing hierarchical self-organizing map (I-GHSOM) for IDSs were proposed in [37]. In [38], the authors presented a transfer learning-based detection system and two update schemes to work efficiently in an environment with a few labeled data. The first scheme is cloud-based, which has very few labeled data, and the second one is for the local environment, which has no labeled data.

3. Machine Learning-Based Intrusion Detection for VANET

To achieve the goal of developing a novel IDS paying more attention to classifying attack cases, the challenges addressed in this study include increasing attention to classifying attack cases correctly and extending the classification of attack cases up to two levels. In this section, the intrusive behavior of a VANET is addressed first, then an innovative ML-based framework is proposed with detailed architecture, and finally, the core of the proposed architecture is discussed.

3.1. Working Procedure of the Proposed Model

This section describes the steps and processes involved in implementing the proposed model. It provides a detailed overview of how the model works, including the various algorithms and techniques used to preprocess the data, train and test the model, and evaluate its performance. According to the problem formulation, we developed a cascaded machine learning-based IDS for VANETs. The experiment was carried out by the following steps:

1. At first, we chose a publicly available dataset (Car Hacking dataset) for our experiments. This is a VANET dataset that contains different types of attacks commonly seen in VANETs.
2. Data were preprocessed in order to conduct the experiment, as there are several unnecessary fields in the dataset that may cause trouble.
3. Then, we developed our proposed model, which has been used throughout the experimentations. This model works in two stages which are described later.
4. Lastly, we carried out a performance evaluation to see the accuracy of our model. We also compared our model with the existing models to justify our work.

3.2. Dataset Processing Overview

Dataset preprocessing is an essential step in any ML project. It involves transforming raw data into a format suitable for analysis and modeling. The goal of dataset preprocessing is to improve the quality and accuracy of the data by removing any noise, inconsistencies, or biases that may affect the results of the analysis. This study uses the Car Hacking dataset [15], which has five classes: Normal, DoS, Gear, RPM, and Fuzzy. The first class is

for benign data, and the rest of the four classes are attacks. A detailed description of the dataset and the preprocessing is given in the next section.

3.3. Cascaded Machine Learning Model Overview

Figure 1 provides an overview of the comprehensive cascaded framework proposed in this study. The initial step involves the division of the available samples into separate training and test sets. The training set is then subjected to the principal component analysis (PCA) algorithm. Meanwhile, the test set is reserved for the subsequent assessment of the performance of the developed system, utilizing a 5-fold cross-validation approach.

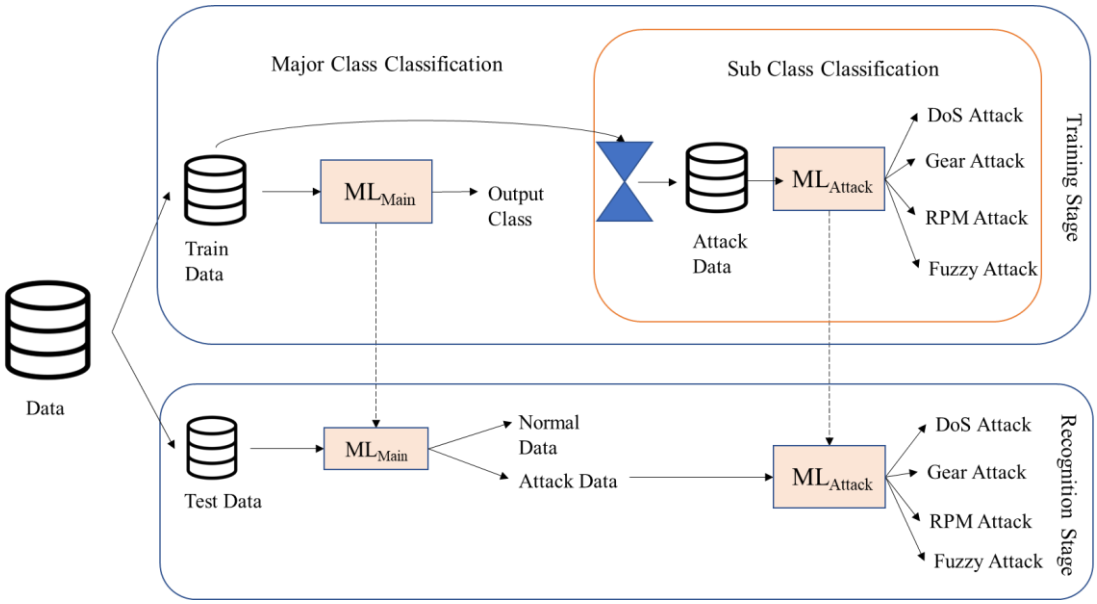

Figure 1. Proposed Cascaded ML IDS Framework in the VANET.

The cascaded framework comprises two distinct stages, each serving a specific purpose. Stage 1 focuses on the initial classification of samples into the major categories of "normal" or "attack". Subsequently, Stage 2 refines the classification process further, particularly for those samples categorized as "attack" during Stage 1.

In Figure 1, the component denoted as "ML_{Main}" within Stage 1 assumes the role of classifying the samples into the broader classes of "normal" or "attack". On the other hand, within Stage 2, the "ML_{Attack}" component comes into play. It specializes in classifying the samples into subclass levels, specifically targeting the four distinct attack subclasses: Denial of Service (DoS), Gear, RPM, and Fuzzy attacks. It is important to note that while the "ML_{Main}" classifier is designed with the potential for two output classes (i.e., "normal" or "attack"), the "ML_{Attack}" classifier accommodates four output classes, corresponding to the aforementioned attack subclasses. This design delineation ensures a refined and accurate classification process, enabling the system to effectively categorize and respond to the various attack scenarios.

The proposed cascaded framework offers a specialized approach for securing VANETs. In the case of the testing phase, a test sample is passed through the ML_{Main} to classify it into a major class, which might be normal or attack. The subclass level of attack-categorized samples is then carried out using the classier trained in Stage 1. For example, if a test sample is classified as an attack, then this attack will be passed to classify its corresponding attack class through the ML_{Attack} model. This method enhances threat detection accuracy and enables targeted response strategies within VANETs, bolstering overall network security despite potential misclassifications.

ML classifier is the basic computational unit in the proposed cascaded framework, as shown in Figure 1. NNs are well-known for classification tasks, and NNs with several

hidden layers are considered classifiers in the proposed cascaded framework. The NN architecture is different in Stage 1 and Stage 2 due to other class numbers to classify.

NN for classification for Stage 1: As per Figure 1, the training data are used to train the major class classification model ML_{Main} to use them in categorizing two major classes. Therefore, the NN for ML_{Main} will have two neurons in the output layer, where individual neurons will represent a particular class. In the input layer, there are a total of nine neurons. As we move deeper into the architecture, the neural network unfolds with three successive hidden layers. The first hidden layer has 50 neurons. Following this, the second hidden layer employs 20 neurons. Finally, the third hidden layer culminates with 10 neurons. The hidden layers employ the ReLU activation function, while the output layer uses sigmoid activation.

NNs for classification for Stage 2: In the case of the Stage 2 classification model, the attack training dataset is used to train the attack model, called the subclass level classification. This will ultimately categorize the respective attacks into subclass levels. The NNs for this stage are similar to the major class model ML_{Main} except for the output layer. The size of the output layer is based on the number of subclasses, and it will be four in this case.

3.4. Significance of This Study

The proposed cascaded model of this study is chosen as it is significantly different from the existing IDS studies for VANETs in several directions, including an emphasis on not classifying attack samples as normal, using a simple ML model with PCA that ensures only vital features contribute to accuracy, and classifying attacks and normal samples through a two-step process. The proposed model is expected to reduce the risk of misclassifying attacks as normal in accordance with improving overall model accuracy, which is vital in this study. Such attention is not seen in previous studies; most of the studies only focused on improving the accuracy of available samples, which is inadequate if we are concerned about the security domain. For example, consider Model A with an accuracy of 99.75%. Model A has good classification accuracy. But only an accuracy measure is insufficient, as a network may perform transactions of 100 million data packets; if the network misclassifies 0.25 million datapoints in a day, among which some are misclassified data, this may cause disastrous consequences. If we improve the model's performance slightly, say from 99.75% to 99.95%, there will still be 0.05 million misclassified datapoints. Therefore, we think of first classifying normal and malicious data and then try to minimize the misclassification of attack data as normal. If the normal data are classified as an attack, this is also a misclassification, but it may not cause vital harm to the system, whereas the reverse may cause disaster.

Several remarkable studies [15,33] have included vast experiments on the VANET dataset in different circumstances. But most of the models have used complex DL models for their experimentation. But this is not needed, as shown in this work, because this dataset is quite balanced and generated in an ideal situation through simulation. Complex models will do better, but if we can use a more generalized architecture with a simple model from a different framework, this approach will reduce the cost and time in the case of complexity in general, as we propose a more simplified model with fewer trainable parameters. For this reason, the proposed model is efficient for saving time and performing better than complex models.

Attack categorization into subclass levels through a cascaded framework is a significant contribution of this study. The Car Hacking dataset (described later) holds five classes; one class is attack-free, i.e., benign data, and the other four classes belong to the attack class. The model at first classifies two major classes where data are classified as an attack or not. After that, if that data are classified as an attack, the subclass classification checks to which attack class it belongs. This two-stage classification increases the accuracy and decreases the possibility of categorizing a normal sample as an attack. The proposed model's perfor-

mance has been evaluated through rigorous experiments for both the Cascaded and Single ML models, presented in the following sections.

4. Experimental Studies

The IDS being proposed is a cascaded framework that incorporates a subclass division model. Additionally, a Single ML model has been examined to substantiate the viability of this framework. To estimate the performance of the proposed model, pioneer and recent works have been considered for comparison. The rest of the section explains the dataset, experimental setup, experimental results, and performance comparisons.

4.1. Dataset Description and Preprocessing

The Car Hacking dataset [15] contains 11 features or attributes and four different attack types. CAN is a state-of-the-art VANET model with much connectivity in the network. Therefore, the dataset was created by sorting CAN traffic where malicious messages were injected into real vehicles. Each interruption was performed for 3 to 5 s in CAN traffic for 30 to 40 min. Table 1 summarizes the features of the Normal and the attack classes.

Table 1. Attributes and Number of Samples in the Car Hacking Dataset.

Attack Names	Timestamp	CAN ID	Number of Data Bytes	Data [0–7]	Output Class	Number of Samples
Normal	--	Random hex value	0–8 (but all are 8)	Data value	T	988,872
Denial of Service (DoS)	Every 0.3 milliseconds	Dominant value "0000"	0–8 (but all are 8)	Data value	R	587,521
Gear attack (spoofing type)	Every 1 millisecond	Random hex value	0–8 (but all are 8)	Data value	R	597,252
RPM attack (spoofing type)	Every 1 millisecond	Random hex value	0–8 (but all are 8)	Data value	R	654,897
Fuzzy attack	Every 0.5 milliseconds	Random hex value	0–8 (but all are 8)	Data value	R	491,847

The dataset contains 5 .csv files for each class. In each .csv file for the attack class, data are available for that class (labeled as "R") as well as the Normal class (labeled as "T"), except the .csv file for the Normal class, which contains solely data for the Normal class. In the preprocessing phase, we took the complete .csv file that contains Normal data, and only the "R" labeled data were taken from the rest of the four attack .csv files, which resulted in total data of 988,872, 587,521, 597,252, 654,897, and 491,847 samples for Normal, DoS, Gear, RPM, and Fuzzy, respectively. All attribute values were numeric. Numeric attributes underwent scaling to the range of 0 to 1.0. For the primary class model and the subclass model, output levels underwent conversion using one-hot encoding. In this scheme, to ensure balanced datasets, available samples were partitioned into training and test sets. For classes with substantial samples, 20% were reserved as test samples, while around 80% of the remaining data constituted the training set. Notably, the dataset's balance was maintained, and the distribution of different attack types is delineated in the subsequent discussion of the confusion matrix within this section.

The 8-byte data features (Data [0–7]) are important to explain. These 8 bytes are set to a constant value for DoS, Gear, and RPM class samples. The variations are present in Fuzzy and Normal data samples, where the min and max values are 0 and 255, respectively, for both classes. The Fuzzy data have an average of around 127 for all 8-byte features, while the Normal data have different average values for their 8-byte features.

Based on the dataset information, classes like DoS, Gear, and RPM have constant values in the 8-byte features, indicating no variation or noise in these aspects. There is variation within the specified value range (0 to 255) for the Fuzzy and Normal data, but this range does not inherently suggest noise. Fuzzy data's consistent average of 127 across all

features suggests coherence while varying averages in Normal data might indicate diversity rather than noise. Overall, the dataset does not immediately appear noisy; variations seem to adhere to predefined ranges and exhibit patterns rather than random noise.

4.2. Experimental Setup

The Python programming language and Anaconda IDE were used for the experiment. The PC on which the experiments were conducted had the following configuration: Processor of 7th Generation Intel® Core™ i5-7400 CPU @ 3.50GHz, GPU of NVIDIA Ge-Force GTX 1070Ti, 8 GB. All the models were trained with Adam optimizer with a learning rate of 0.001 for 300 epochs. The architecture of NNs for Cascaded ML is explained as the proposed model; the NN architectures of both models are the same. The NN architecture has been discussed in the previous section.

To evaluate the overall performance of the two implemented models, the precision, recall, $F1$ score, and accuracy of the models are measured on the test samples. Equations (1)–(4) are the well-known measures of the scores.

$$Precision = TP/(TP + FP) \tag{1}$$

$$Recall = TP/(TP + FN) \tag{2}$$

$$F1\ Score = 2 \times (Precision \times Recall)/(Precision + Recall) \tag{3}$$

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{4}$$

where TP stands for true positive, which means the model correctly predicted the positive class. FP stands for false positive, meaning the model predicted the positive class, but it was not true. In other words, the model made an incorrect positive prediction. FN is for false negative, meaning that the model predicted the negative class (e.g., no disease), but it was actually true that the positive class was present. In other words, the model missed a positive instance.

Precision is a metric that measures the accuracy of positive predictions made by a model. It is calculated as the ratio of true positives (correctly predicted positive cases) to the sum of true positives and false positives (incorrectly predicted positive cases). Precision helps to assess the reliability of positive predictions and is particularly important in scenarios where false positives are costly or undesirable.

Recall is a metric that measures the model's ability to identify all relevant instances within a dataset. It is calculated as the ratio of true positives to the sum of true positives and false negatives (cases that were actually positive but were predicted as negative). Recall is important when missing positive instances, such as medical diagnoses, is a critical concern.

The $F1$ score is a harmonic mean of precision and recall. It provides a balanced evaluation of a model's performance by considering both false positives and false negatives. The formula for calculating the $F1$ score is shown in Equation (3). The $F1$ score is useful when there is a need to balance precision and recall, as it combines both aspects into a single metric.

The performance of implemented models was also analyzed through a confusion matrix. A confusion matrix is a tabular representation to evaluate the performance of a classification model. It presents the comparison of predicted and actual class labels for a set of datapoints through TP, TN, FP, and FN. These values provide insights into the model's accuracy and error rates.

4.3. Experimental Results and Analysis

This section presents experimental outcomes from the Car Hacking dataset with two models. Insightful evaluation of model-wise misclassification is the main/crucial point of

the study to realize the proposed approach. Finally, the overall performances of the models are analyzed, and then the outcome of the proposed method is compared with existing methods.

Figures 2 and 3 show the loss and accuracy curves of the two different implemented models with NNs in a single and cascaded manner. For simplicity, the cascaded model's curve is for NNs of ML_{Main} in Stage 1. Therefore, the curve for the cascaded case is for major class classification. It is visible from the loss curve that the Single ML model holds the highest amount of loss. On the other hand, it is clearly seen that the Cascaded ML model has the lowest amount of loss. In addition, the Cascaded ML model has the highest accuracy, as shown in Figure 2. At a glance, the Cascaded ML model is much more efficient than the other models, and its proficiency is explained in detail with the help of bar charts with confusion matrices.

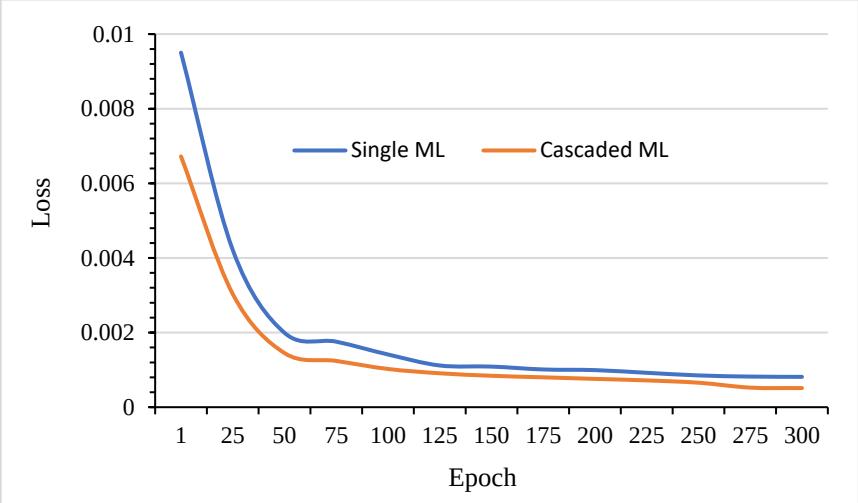

Figure 2. Loss Curves of the Two Implemented Methods: Single ML and proposed Cascaded ML.

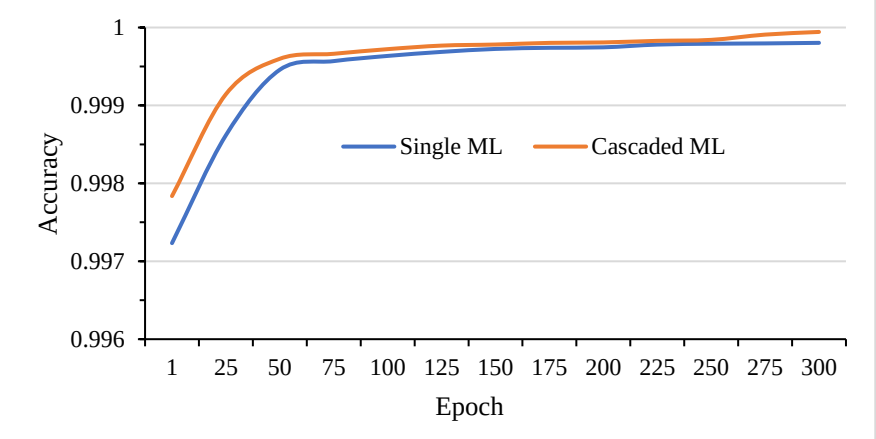

Figure 3. Accuracy Curves of the Two Implemented Methods: Single ML and proposed Cascaded ML.

Table 2 represents the test set confusion matrix of the SINGLE ML model with a misclassification summary. For the Normal class, there were 185,628 test samples, in which four attacks were misclassified as Fuzzy, and one attack was misclassified as RPM. DoS data had 117,505 samples, and all of them were classified correctly. There were 98,370 samples in the Fuzzy attack class, of which 85 samples went into the Normal class and 8 samples went into the RPM class. Gear and RPM had 119,451 and 130,978 samples, respectively, classified correctly. So, it is clear that the Single ML model can classify data correctly, and other complex DL models (LSTM, CNN, etc.) are not needed for this dataset. In short, a total of 98 errors were noticed, and 85 of them were grouped into the Normal class. One

reason behind this is that the dataset is quite balanced, and the values of the attributes were taken in an ideal environment. So, our Single ML model with a different framework worked fine in this case.

Table 2. Confusion Matrix for Single ML Model.

Attacks (Total Samples)	Normal	DoS	Fuzzy	Gear	RPM	Misclassification Summary	
						As Normal	Other Classes
Normal (185,628)	185,623	0	4	0	1	0	5
DoS (117,505)	0	117,505	0	0	0	0	0
Fuzzy (98,370)	85	0	98,277	0	8	85	8
Gear (119,451)	0	0	0	119,451	0	0	0
RPM (130,978)	0	0	0	0	130,978	0	0
					Total	85	13

Table 3 represents the test set confusion matrix for the proposed Cascaded ML model with a misclassification summary, which is crucial to realize proficiency of the proposed model. The test samples were the same as for the Single ML model. For the Normal class, no attack was misclassified, whereas for the Single ML model (Table 2), the misclassifications were in two different classes. No DoS, Gear, or RPM attacks were classified as Normal or into other classes, which means the proposed model was able to classify these three classes correctly. For Fuzzy data, 61 attack data went into the Normal class, but the number was 85 for the Single ML model, shown in Table 2. Our major goal, which is reducing the misclassification of attacks as normal data, is also achieved by the Cascaded ML model. More significantly, no other misclassification happened for other classes. Overall, the total misclassification of 61 samples revealed the outperformance of the Cascaded ML model over the Single ML model, which misclassified 98 (=85 + 13) samples, as seen in Table 2. Consequently, our proposed Cascaded ML model with a different framework worked fine and provided promising outcomes.

Table 3. Confusion Matrix for Cascaded ML Model.

Attacks (Total Samples)	Normal	DoS	Fuzzy	Gear	RPM	Misclassification Summary	
						As Normal	Other Classes
Normal (185,628)	185,628	0	0	0	0	0	0
DoS (117,505)	0	117,505	0	0	0	0	0
Fuzzy (98,370)	61	0	98,309	0	0	61	0
Gear (119,451)	0	0	0	119,451	0	0	0
RPM (130,978)	0	0	0	0	130,978	0	0
					Total	61	0

The summary of the experiment analyzed above is presented in Table 4. The table recaps the overall performance of the two models in the sense of precision, recall, *F*1 score, and accuracy. According to the results presented in the table, our proposed cascaded model performs better than the Single ML model.

Table 4. Overall Performance Comparison between the Two Implemented Models: Single ML and proposed Cascaded ML.

Model	Precision (%)	Recall (%)	F1 Score (%)	Accuracy (%)
Single ML Model	99.98	99.98	99.98	99.98
Cascaded ML Model	99.98	99.99	99.98	99.99

Table 5 illustrates the accuracy of the Single ML model according to PCA, and the test set accuracy is computed with 5-fold cross-validation. It is clear that for certain folds, the

Single ML model can reach up to 100% accuracy, but this is not true for every fold. Thus, the average accuracy is considered, which is highest when nine features are considered. Table 6 demonstrates the accuracy of the Cascaded ML model, showing ML_{Main} and ML_{Attack} individually. It is evident that ML_{Main} can reach accuracy of 100% when the number of features is five for all 5-fold test data, and the same is true for ML_{Attack} with four features.

Table 5. Accuracy of the Single ML Model with Features Selected by PCA.

No. of Features	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg.: 5-Fold CV
1	0.9745	0.9693	0.9769	0.9739	0.9747	0.97386
2	0.9925	0.9942	0.9918	0.9905	0.9951	0.99282
3	0.9979	0.998	0.9983	0.9973	0.9971	0.99772
4	0.9999	0.9998	0.9999	1	0.9999	0.9999
5	1	1	0.9999	0.9999	0.9999	0.99994
6	1	0.9999	0.9999	0.9999	0.9999	0.99992
7	1	0.9999	0.9999	0.9999	1	0.99994
8	1	0.9999	0.9999	0.9999	0.9999	0.99992
9	1	1	1	0.9999	0.9999	0.99996
10	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
11	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998

Table 6. Accuracy of the Cascaded ML model with Features Selected by PCA.

	ML_{Main}						ML_{Attack}					
No. of Features	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg.: 5-Fold CV	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg.: 5-Fold CV
1	0.9772	0.9729	0.9795	0.9779	0.9777	0.97704	0.9889	0.9874	0.9852	0.9881	0.9899	0.9879
2	0.9925	0.9925	0.993	0.9924	0.9923	0.99254	0.9999	0.9999	0.9999	0.9999	0.9999	0.9999
3	0.9979	0.9973	0.9969	0.9981	0.9976	0.99756	1	1	0.9999	0.9999	0.9999	0.99994
4	1	1	0.9999	0.9999	1	0.99996	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1

Figure 4 summarizes the results presented in Tables 5 and 6. It can be concluded from the figure that the proposed cascaded framework (the Cascaded ML model) reaches a maximum classification accuracy of 100% when PCA is 5, but the same is not true for a simple NN (the Single ML model). That solidifies the necessity of our proposed framework.

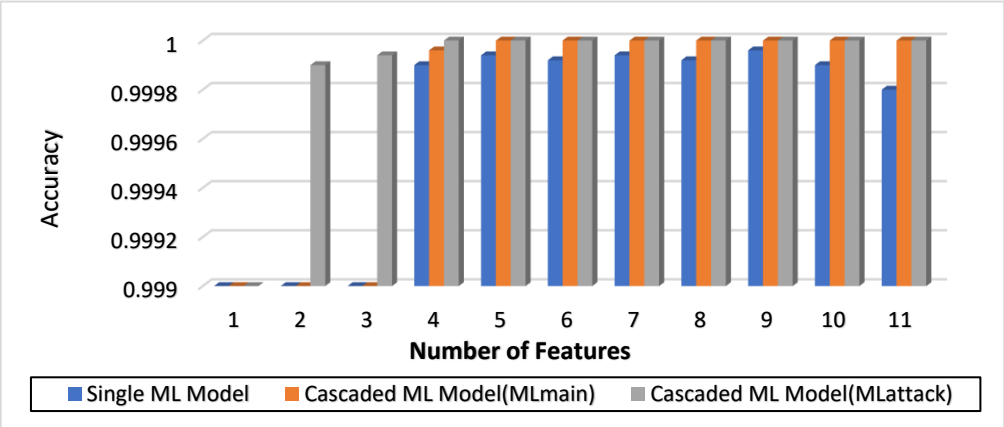

Figure 4. Accuracy Chart of Different Models.

4.4. Performance Comparison with Existing Models

Table 7 compares the performance of the proposed IDS with other prominent works with the Car Hacking dataset and other generated or simulated datasets. Computational methods used by the individual studies and other important or remarkable issues are mentioned in the table for better comparison. As stated earlier, the previous IDS works on VANETs focused on increasing the accuracy of the models. Additionally, most models were complex under the framework to improve accuracy. But the focus of this study was not only increasing the model's accuracy but also giving more attention to handling attack samples to ensure they are not classified as normal.

Moreover, this study aimed to employ the Single ML model under a completely different framework for handling malicious attacks. The efforts eventually helped to reach a satisfactory accuracy of the proposed model. Finally, the proposed Cascaded ML model (i.e., the proposed model) achieved remarkable accuracy of 99.99%, the best among the methods mentioned in the table in the sense of using a different framework with the NN architecture in the core. Furthermore, the proposed models' evaluation was carried out in two stages where the model first classifies the attack and normal data and then categorizes the attack data into different subclasses with the help of a cascaded model. This process helped to achieve our desired goal as well as generated praiseworthy results without deploying any complex models. Not only our proposed model but also the first model worked fine in the environment. The challenging issue is to design a cost-effective and effective model without using any complexity in the model. A significant breakthrough of our approach lies in its departure from employing resource-intensive models such as deep CNNs, autoencoders, and LSTM autoencoders. Instead, we harness the power of a simplified NN, remarkably outperforming existing benchmarks by emphasizing attack data not being classified as normal data. This strategic utilization of a leaner architecture underscores our framework's efficacy in achieving superior results. So, the proposed method is much more effective than other existing methods.

Table 7. Performance Comparison of the Proposed Model with State-of-the-Art Models.

Work Ref.	Method	Dataset	Test Data Split	Accuracy	Level of Classification	Attack Emphasized	Remarks and Significance
Song et al., 2020 [15]	Deep CNN	Generated dataset	30% of the available data	Not mentioned	Major class only	No	Importance of the accuracy of the DL model
Hossain et al., 2020 [34]	LSTM	Generated dataset	Not mentioned	99.99%	Major class only	No	Importance of the accuracy of the DL models
Barletta et al., 2020 [28]	Unsupervised Kohonen self-organizing map	Car Hacking dataset [15]	20% of the available data	99.88%	Major class only	No	Emphasis on the accuracy of the ML model
Ashraf et al., 2021 [35]	LSTM autoencoder	Car Hacking dataset [15]	20% of the available data	99%	Major class only	No	Importance of the accuracy of the DL models
		UNSW-NB15 [39]	20% of the available data	98%			
Ullah et al., 2022 [33]	Hybrid DL model (LSTM + GRU)	Car Hacking [15] and DDoS dataset [17]	20% of the available data	99.70%	Major class only	No	Importance of the accuracy of the DL model
Proposed Model	Single ML model	Car Hacking dataset	20% of the available data	99.996%	Classification of attacks and major attack classes	Yes	Cascaded framework and importance of the number of attacks classified as normal
	Cascaded ML model			100%			

5. Conclusions

This work introduces a machine learning-based IDS for VANETs that effectively classifies attacks into subclasses while preventing the misclassification of attack classes as the Normal class and improving overall accuracy. The proposed Cascaded ML model, implemented using a simple NN architecture and incorporating PCA, demonstrates superior performance compared to prevailing complex models, making it easier to understand and implement. Experimental results on the Car Hacking dataset have revealed that the proposed model performs what is desired for this study: The proposed cascaded model efficiently reduced the attack class misclassification as normal from 85 to 61, which is crucial in defending a system from suspicious attacks and a major concern of the study. Finally, based on superior accuracy, the proposed method achieved 100% accuracy when coupled with PCA. Therefore, CascadMLIDS is identified as an effective IDS with distinct properties to tackle attack issues concerning the existing state-of-the-art ML-based IDS models.

Several future potential research directions can be explored based on this study. Firstly, other approaches, like the game theoretic approach, might perform differently instead of NNs in the proposed cascaded framework. Another future work could be to adopt this IDS for a real-time scenario in an underlying security system. More interestingly, the idea of data augmentation and cascaded frameworks will be applicable for similar scenario cases, such as security issue handling and preventing an attacker from getting into the network in a VANET, since this will also be challenging due to the lack of a strong and protected system in VANETs.

Author Contributions: Conceptualization, M.A.H.A.; Validation, M.A.S.K.; Formal analysis and software, A.C.D. and A.R.; Writing—original draft, A.C.D.; Writing—review & editing, A.R., M.A.H.A. and M.A.S.K.; Supervision, M.A.H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Freely available Car Hacking dataset (https://ocslab.hksecurity.net/Datasets/car-hacking-dataset) is used in this study. Related materials of this study are available in GitHub repository (https://github.com/dhar7/CascadMLIDS).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ghori, M.R.; Zamli, K.Z.; Quosthoni, N.; Hisyam, M.; Montaser, M. Vehicular ad-hoc network (VANET): Review. In Proceedings of the 2018 IEEE International Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 11–12 May 2018; pp. 1–6. [CrossRef]
2. Azam, F.; Yadav, S.K.; Priyadarshi, N.; Padmanaban, S.; Bansal, R.C. A Comprehensive Review of Authentication Schemes in Vehicular Ad-Hoc Network. *IEEE Access* **2021**, *9*, 31309–31321. [CrossRef]
3. Al Junaid, M.A.H.; Syed, A.A.; Warip, M.N.M.; Azir, K.N.F.K.; Romli, N.H. Classification of Security Attacks in VANET: A Review of Requirements and Perspectives. *MATEC Web Conf.* **2018**, *150*, 06038. [CrossRef]
4. Mishra, R.; Singh, A.; Kumar, R. VANET security: Issues, challenges and solutions. In Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 1050–1055. [CrossRef]
5. Li, Y.; Liu, Q. A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments. *Energy Rep.* **2021**, *7*, 8176–8186. [CrossRef]
6. Gunduz, M.Z.; Das, R. Cyber-security on smart grid: Threats and potential solutions. *Comput. Netw.* **2020**, *169*, 107094. [CrossRef]
7. Kurniabudi; Stiawan, D.; Darmawijoyo; Idris, M.Y.B.; Defit, S.; Triana, Y.S.; Budiarto, R. Improvement of attack detection performance on the internet of things with PSO-search and random forest. *J. Comput. Sci.* **2022**, *64*, 101833. [CrossRef]
8. Yıldırım, M.; Mackie, I. Encouraging users to improve password security and memorability. *Int. J. Inf. Secur.* **2019**, *18*, 741–759. [CrossRef]
9. Awasthi, A.; Goel, N. Phishing website prediction using base and ensemble classifier techniques with cross-validation. *Cybersecurity* **2022**, *5*, 22. [CrossRef]

10. Aljawarneh, S.; Aldwairi, M.; Yassein, M.B. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *J. Comput. Sci.* **2018**, *25*, 152–160. [CrossRef]
11. Ahmad, Z.; Khan, A.S.; Shiang, C.W.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [CrossRef]
12. Arshad, J.; Azad, M.A.; Amad, R.; Salah, K.; Alazab, M.; Iqbal, R. A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT. *Electronics* **2020**, *9*, 629. [CrossRef]
13. Pharate, A.; Bhat, H.; Shilimkar, V.; Mhetre, N. Classification of Intrusion Detection System. *Int. J. Comput. Appl.* **2015**, *118*, 23–26. [CrossRef]
14. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
15. Song, H.M.; Woo, J.; Kim, H.K. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **2020**, *21*, 100198. [CrossRef]
16. Seo, E.; Song, H.M.; Kim, H.K. GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland, 28–30 August 2018; pp. 1–6. [CrossRef]
17. Kang, M.-J.; Kang, J.-W. Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. *PLoS ONE* **2016**, *11*, e0155781. [CrossRef] [PubMed]
18. Ludwig, S.A. Applying a Neural Network Ensemble to Intrusion Detection. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 177–188. [CrossRef]
19. Sarkar, S.K.; Basavaraju, T.G.; Puttamadappa, C. *Ad Hoc Mobile Wireless Networks*; CRC Press: Boca Raton, FL, USA, 2016. [CrossRef]
20. Hasrouny, H.; Samhat, A.E.; Bassil, C.; Laouiti, A. VANet security challenges and solutions: A survey. *Veh. Commun.* **2017**, *7*, 7–20. [CrossRef]
21. Silva, T.H.; Celes, C.S.F.S.; Neto, J.; Mota, V.; Cunha, F.; Ferreira, A.; Ribeiro, A.I.J.T.; Vaz de Melo, P.; Almeida, J.; Loureiro, A. Users in the urban sensing process. In *Pervasive Computing*; Elsevier: Amsterdam, The Netherlands, 2016; pp. 45–95. [CrossRef]
22. Paul, A.; Chilamkurti, N.; Daniel, A.; Rho, S. Introduction: Intelligent vehicular communications. In *Intelligent Vehicular Networks and Communications*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 1–20. [CrossRef]
23. Al-Heety, O.S.; Zakaria, Z.; Ismail, M.; Shakir, M.M.; Alani, S.; Alsariera, H. A Comprehensive Survey: Benefits, Services, Recent Works, Challenges, Security, and Use Cases for SDN-VANET. *IEEE Access* **2020**, *8*, 91028–91047. [CrossRef]
24. Mahmood, J.; Duan, Z.; Yang, Y.; Wang, Q.; Nebhen, J.; Bhutta, M.N.M. Security in Vehicular Ad Hoc Networks: Challenges and Countermeasures. *Secur. Commun. Netw.* **2021**, *2021*, 9997771. [CrossRef]
25. Quyoom, A.; Mir, A.A.; Sarwar, D.A. Security Attacks and Challenges of VANETs: A Literature Survey. *J. Multimed. Inf. Syst.* **2020**, *7*, 45–54. [CrossRef]
26. Talpur, A.; Gurusamy, M. Machine Learning for Security in Vehicular Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 346–379. [CrossRef]
27. Nazakat, I.; Khurshid, K. Intrusion Detection System for In-Vehicular Communication. In Proceedings of the 2019 15th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2–3 December 2019; pp. 1–6. [CrossRef]
28. Barletta, V.S.; Caivano, D.; Nannavecchia, A.; Scalera, M. Intrusion Detection for in-Vehicle Communication Networks: An Unsupervised Kohonen SOM Approach. *Future Internet* **2020**, *12*, 119. [CrossRef]
29. Ingre, B.; Yadav, A.; Soni, A.K. Decision tree based intrusion detection system for NSL-KDD dataset. In *Smart Innovation, Systems and Technologies*; Springer: Cham, Switzerland, 2018. [CrossRef]
30. Junejo, M.H.; Rahman, A.A.-H.A.; Shaikh, R.A.; Yusof, K.M.; Kumar, D.; Memon, I. Lightweight Trust Model with Machine Learning scheme for secure privacy in VANET. *Procedia Comput. Sci.* **2021**, *194*, 45–59. [CrossRef]
31. Haydari, A.; Yilmaz, Y. RSU-Based Online Intrusion Detection and Mitigation for VANET. *Sensors* **2022**, *22*, 7612. [CrossRef] [PubMed]
32. Bangui, H.; Ge, M.; Buhnova, B. A Hybrid Data-driven Model for Intrusion Detection in VANET. *Procedia Comput. Sci.* **2021**, *184*, 516–523. [CrossRef]
33. Ullah, S.; Khan, M.A.; Ahmad, J.; Jamal, S.S.; e Huma, Z.; Hassan, M.T.; Pitropakis, N.; Arshad; Buchanan, W.J. HDL-IDS: A Hybrid Deep Learning Architecture for Intrusion Detection in the Internet of Vehicles. *Sensors* **2022**, *22*, 1340. [CrossRef]
34. Hossain, M.D.; Inoue, H.; Ochiai, H.; Fall, D.; Kadobayashi, Y. LSTM-Based Intrusion Detection System for In-Vehicle Can Bus Communications. *IEEE Access* **2020**, *8*, 185489–185502. [CrossRef]
35. Ashraf, J.; Bakhshi, A.D.; Moustafa, N.; Khurshid, H.; Javed, A.; Beheshti, A. Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events from Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 4507–4518. [CrossRef]
36. Nie, L.; Li, Y.; Kong, X. Spatio-Temporal Network Traffic Estimation and Anomaly Detection Based on Convolutional Neural Network in Vehicular Ad-Hoc Networks. *IEEE Access* **2018**, *6*, 40168–40176. [CrossRef]
37. Liang, J.; Chen, J.; Zhu, Y.; Yu, R. A novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) based on differences of traffic flow and position. *Appl. Soft Comput.* **2019**, *75*, 712–727. [CrossRef]

38. Li, X.; Hu, Z.; Xu, M.; Wang, Y.; Ma, J. Transfer learning based intrusion detection scheme for Internet of vehicles. *Inf. Sci.* **2021**, *547*, 119–135. [CrossRef]
39. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015. [CrossRef]