




Article

HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System

Emad Ul Haq Qazi , Muhammad Hamza Faheem  and Tanveer Zia 

Center of Excellence in Cybercrimes and Digital Forensics (CoECDF), Naif Arab University for Security Sciences (NAUSS), Riyadh 14812, Saudi Arabia

* Correspondence: qabdulrab@nauss.edu.sa

Abstract: Attacks on networks are currently the most pressing issue confronting modern society. Network risks affect all networks, from small to large. An intrusion detection system must be present for detecting and mitigating hostile attacks inside networks. Machine Learning and Deep Learning are currently used in several sectors, particularly the security of information, to design efficient intrusion detection systems. These systems can quickly and accurately identify threats. However, because malicious threats emerge and evolve regularly, networks need an advanced security solution. Hence, building an intrusion detection system that is both effective and intelligent is one of the most cognizant research issues. There are several public datasets available for research on intrusion detection. Because of the complexity of attacks and the continually evolving detection of an attack method, publicly available intrusion databases must be updated frequently. A convolutional recurrent neural network is employed in this study to construct a deep-learning-based hybrid intrusion detection system that detects attacks over a network. To boost the efficiency of the intrusion detection system and predictability, the convolutional neural network performs the convolution to collect local features, while a deep-layered recurrent neural network extracts the features in the proposed Hybrid Deep-Learning-Based Network Intrusion Detection System (HDLNIDS). Experiments are conducted using publicly accessible benchmark CICIDS-2018 data, to determine the effectiveness of the proposed system. The findings of the research demonstrate that the proposed HDLNIDS outperforms current intrusion detection approaches with an average accuracy of 98.90% in detecting malicious attacks.

Keywords: intrusion detection; CICIDS-2018; deep learning; convolution neural networks; recurrent neural networks



Citation: Qazi, E.U.H.; Faheem, M.H.; Zia, T. HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System. *Appl. Sci.* **2023**, *13*, 4921. <https://doi.org/10.3390/app13084921>

Academic Editor: Gianluca Lax

Received: 17 January 2023

Revised: 10 April 2023

Accepted: 11 April 2023

Published: 14 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Functions of information and communication technology (ICT) systems are critical in all aspects of the industry and human life. In recent decades, numerous organizations have become vulnerable to sophisticated cyber-attacks, resulting in the formation of a revolutionary Intrusion Detection System (IDS). An IDS is an unutilized network security method for identifying various forms of malicious intrusions. John Anderson was the first person to put in a significant amount of work in the identification field in the year 1980 [1]. Every cyber-attacks entail economic costs, reputational harm, and legal consequences; hence, the development of IDSs has a global impact on both the academic community and the business sector.

It is important that networks be protected against unwanted access, and that user engagement and user data be safeguarded [2], in addition to revealing new security vulnerabilities. An intrusion detection system is an effective security-enhancing technique for identifying and preventing networks or systems from cyber assaults. IDSs are responsible for the identification of suspicious activities and the overall security of a network infrastructure against cyberattacks and for reducing financial and operational losses [3]. According to the literature, a network architecture determines the classification of IDSs according to three categories:

- Intrusion detection systems based on the network [4], which examine the components of unique packets to detect harmful network traffic behavior patterns.
- Server signature IDS [5], which analyzes the activity system logs of individual hosts and identifies malicious attacks and hybrid identification systems [6].
- Systems that use anomaly and signature-based intrusion detection systems have higher quality and stronger security practices.

The signature detection method makes use of predetermined patterns and classifiers to better assess malicious assaults. It utilizes existing information to identify harmful threats; hence, it is named a strategy based on knowledge. The approach achieves a low false positive (FP) along with higher accuracy, but it is incapable of detecting new network attacks [7]. To discover unknown hostile threats, the anomaly detection approach employs heuristic methods. As a result, the effectiveness of this anomaly detection approach for detecting anomalies is good despite a high false-positive rate. Various businesses have adopted protocol analysis, which uses a combination of anomaly and signature-based systems, to avoid this problem [8]. According to the deployment pattern, ID systems are categorized into two main types, which are distributed and non-distributed. A distributed implementation consists of several ID subsystems connected over a vast network, whereas a non-distributed structure, such as an open-source snort, may be deployed in a specific location [9].

In modern days, current methods to detect network intrusions in industries include statistical testing and threshold computation techniques. The ID system based on statistical tests relies on numerous traffic limitations, such as packet length, the timing of packet arrival, and traffic flow volume, depending on the network traffic of the model in a predetermined amount of time. Due to the complexity of today's modern malicious attacks, it is possible that these strategies will not be effective. In replacement of these statistically based techniques, a solution that is most optimized and efficient is needed. Machine learning (ML)-based approaches have been widely utilized to help network administrators deal with a wide range of harmful attacks in preventing these attacks [10].

Ensemble learning (EL) enhances ML outcomes by combining many models into one. These algorithms evaluate the state of the network by classifying the processed data into normal and abnormal categories. These algorithms exercise and simulate attacks with precision to evaluate capabilities with diverse datasets. Nevertheless, most of these datasets are extremely imbalanced. A total of 98% of these datasets are regarded as normal, whereas the remaining 2% are categorized as assaults [11]. Folino et al. [12] suggested a novel deep-learning model based on ensemble learning for interpreting non-stationary datasets such as IDS logs. It is desirable to be able to construct a better detection system, especially when utilizing ensemble classifiers. When creating an ensemble, selecting suitable classifiers and deciding on combiners are two important issues. In [13], a study of ensemble learning for IDSs was provided by Tama. However, most traditional ML techniques fall within the area of superficial learning and place a low emphasis on the design and selection of features; they are incapable of addressing the large classification task imposed by attack data in a real-world network application. As the number of datasets continues to grow, the accuracy of multiclassification attack detection will decrease. Consequently, intelligent assessment is inconsistent with ML and the projection requirements of higher-dimensional learning with vast amounts of data [14].

This study aimed to develop a network intrusion detection system that is based on flow-based statistics utilizing the benchmark Canadian Institute for Cybersecurity intrusion detection system (CICIDS) 2018 dataset, which accurately identifies and categorizes every type of attack using a multi-categorization scheme. To identify network traffic, we developed an improved 1D CNN-based deep neural network model. The main contributions of this paper are as follows.

- Proposal of a deep-layered architecture using the recurrent neural network (RNN) and convolutional neural network (CNN) to detect and classify malicious traffic.
- Detailed analysis of existing machine learning and deep learning techniques.

- In-depth analysis of the CICIDS 2018 dataset.

The remaining sections of this work are structured as follows: Section 2 presents the background while Section 3 explains the related work. Section 4 discusses the proposed HDLNIDS model. The dataset explanation is presented in Section 5. The experimental details and discussion are presented in Section 6, followed by a comparison with existing approaches in Section 7, and finally the summary and the conclusion in Section 8.

2. Background

This section covers the details about current IDSs and existing deep-learning-based methodologies for their detection.

2.1. Intrusion Detection Systems

The use of network monitoring in forensics, security, and anomaly detection has become commonplace. However, recent developments have introduced several additional challenges for IDSs. The most relevant concerns are as follows.

2.1.1. Volume

Data are being stored and sent over networks in ever-increasing amounts. The amount of data that was accessible in 2020 was predicted to exceed 44 ZB. By 2025, the amount of data generated each day is expected to reach 463 exabytes globally [4]. As a result, modern networks' traffic capacity has increased dramatically to accommodate the observed traffic level. Numerous contemporary backbone connections currently operate at wire speeds of 100 Gbps or more. To put this into context, a 100 Gbps network can handle 148,809,524 packets per second (PPS) [5]. To function at wire speeds, an IDS must be capable of analyzing within 6.72 nanoseconds. Providing an IDS at such a rapid rate is challenging, and achieving adequate efficacy, accuracy, and efficiency is similarly a challenge.

2.1.2. Accuracy

Existing methods cannot be depended upon to maintain adequate accuracy. To give a more thorough and accurate viewpoint, higher levels of precision, depth, and contextual knowledge are required [5]. Unfortunately, this imposes several financial, computational, and time-related constraints.

2.1.3. Diversity

In recent years, there has been a rise in the number of novel or customized protocols used in contemporary networks [5]. This is largely attributable to the amount of network and/or internet-connected gadgets. Consequently, it is becoming more difficult to distinguish between regular and anomalous traffic behavior.

2.1.4. Dynamics

Due to the complexity and adaptability of contemporary networks, their behavior is dynamic and difficult to anticipate [5]. In turn, this makes it impossible to develop a dependable behavioral standard. It also raises questions regarding the longevity of learning models.

2.1.5. Low-Frequency Attacks

Based on the complex behavior of contemporary networks, different types of attacks have frequently defeated earlier systems for detecting anomalies, including artificial-intelligence-based approaches [6]. As a result of imbalances in the training dataset, an IDS delivers less precise detection when confronted with low-frequency attacks.

2.1.6. Adaptability

Inadequate precision, dynamic network traffic behavior, low-frequency network attacks, flexibility to software-defined networks, the enormous volume of stored and sent

data, and a variety of network access devices are significant obstacles for modern NIDSs. Modern networks have embraced several new technologies to decrease their dependency on static technology and management techniques [6]. Therefore, dynamic technologies such as containerization, virtualization, and Software-Defined Networks (SDNs) are being utilized. IDSs will need to adapt to the use of these technologies and the adverse effects they produce.

2.2. Deep-Learning-Based IDS

Deep learning is a subfield of machine learning. Using several layers of representation helps the modeling of intricate relationships and concepts [6]. Using the output characteristics of lower levels, supervised and unsupervised learning algorithms are utilized to generate increasingly higher levels of abstraction.

IDSs play an important part in cybersecurity as they defend the network from cyber-attacks by monitoring the network. IDSs in cybersecurity have evolved using deep learning (DL) due to their findings in computer vision, image processing, and natural language processing [15]. Due to their two key properties, hierarchical feature representations and the acquisition of long-term temporal patterning, this structure of hierarchical and heuristic search is highly effective. DL is popular among researchers. Therefore, considerable thought has been given to DL approaches for enhancing the intelligence of IDSs, despite a lack of research comparing such machine learning methods with openly available datasets. DL's complex structuring architecture facilitates high-quality learning for complex data processing. Rapid progress in parallel processing technology has produced a robust system basis for DL approaches. Common prevalent issues with existing ML-based models are as follows: (1) such models do have a false positive rate (FP) with a wider variety of malicious invasions [16]; (2) such proposed models are not able to generalize, because most available detection systems skip novel attack vectors because of obsolete ID datasets; (3) better solutions are required to sustain today's rapidly expanding rising internet traffic in a heterogeneous network.

The Canadian Institute for Cybersecurity intrusion detection system (CICIDS) 2018 dataset, which is an upgrade to CICIDS 2017, is frequently utilized. One of the primary reasons for its consideration is that it was developed to address the issues observed in its earlier CICIDS 2017 dataset [17]. It also comprises various kinds of traffic and real-world network traffic, which is one of the primary reasons for its popularity [18]. Although CICIDS 2018 is a suitable dataset, there is a significant issue that must be addressed. The issue's impacts result in a high-class imbalance that directly misleads the classifier [19].

3. Related Work

Over recent decades, ML and DL approaches have been widely utilized regarding the security of the network due to their capacity to distinguish data [18,20–22]. Previously, researchers have employed a variety of ML- and DL-based techniques for ID. Using the KDDCUP ID dataset, Xu et al. [23] used the K-Nearest Neighbor (KNN) for the identification of network anomalies and assessed the effectiveness of the suggested ID system. Bhati et al. [24] used different versions of support vector machine (SVM), including quadratic and linear Gaussians, to evaluate the efficiency of SVM approaches on the NSL-KDD dataset. Sumaiya et al. [25] proposed an integrated ID system employing correlation-based feature selection and the artificial neural network (ANN). Using the datasets of UNSW-NB15 and NSL-KDD ID, the authors conducted an experimental study. Waskle et al. [26] proposed a Random Forest (RF)-based ID system, while Alqahtan et al. [27] proposed a system for identification based on multiple conventional machine learning classification algorithms. Prior methodologies applied within the scope of ID, however, had inefficient classification effectiveness, with a greater FP and a poor detection rate (DR) in the ID system. Utilizing non-symmetric deep auto-encoder for network intrusion detection problem, Qazi et al. [28] conducted the experiments using the benchmark dataset KDD CUP'99. In another study, a one-dimensional convolutional neural network (1D-CNN) based deep learning system

was proposed by the authors [29] for network intrusion detection. The authors used the benchmark CICIDS2017 dataset for conducting the experiments, while Ahmad et al. [30] proposed network intrusion detection and classification system using AdaBoost-based approach. The authors used a UNSW-NB 15 dataset for network anomaly detection. The experimental findings showed that proposed method effectively detects different forms of network intrusions on computer networks.

Deep learning is a subfield of machine learning consisting of concealed layers to determine the features of the deep network. These approaches are more successful than ML [24] owing to their comprehensive structure and capacity to grasp the relevant aspects of the dataset independently and provide an output. Recently, DL has gained popularity and is being applied for ID; studies indicate that DL outperforms traditional methods. Girdler et al. [31] employed the DL technique for the anomaly based on flow identification developed on a deep neural network (DNN), and the results of experiments demonstrated that DL may be utilized for anomaly identification in networks.

Idhammad et al. [21] proposed a decentralized intrusion detection solution for cloud environments. First, the Naive Bayes model was applied to identify anomalies for data preprocessing; subsequently, for multi-classification, RF was used to determine the pattern of each attack. Using the CICDDS-001 dataset, experiments were carried out using variables such as false-positive rate (FPR) and precision.

Anand et al. [32] introduced an IDS composed of multiple vector classifiers for wireless mesh support. Utilizing genetic-algorithm-based feature selection and SVM classification, the authors chose particular traits to boost efficiency. The system was evaluated using a WMN-generated intrusion dataset and was simulated using a standard intrusion dataset in the network simulator-3 (NS3) simulator. The CICIDS 2017 intrusion dataset was used to assess the model.

Ran et al. [33] proposed an ensemble-based approach for network anomaly identification in an intrusion detection system. This method utilizes a combination of learning and forecasting mechanisms to classify anomalies into various classes. Initially, researchers employed the ANOVA F-test considering the strategy of univariate feature selection [34,35] to determine the performance of features and the link between class labels and data characteristics. In addition, they utilized an automated machine learning model for learning and a Kalman filter for prediction. They utilized a Bayesian optimizer as the optimizer for neural network architecture search (NAS), which finds the most accurate architecture from a list of architectures. This ensemble technique employs a voting mechanism that rates the predictions of both algorithms depending on their average accuracy. On publicly accessible CICIDS-2017 and UNSW-NB15 datasets, they tested the performance of the suggested approach and obtained 97.02 and 98.801 percent accuracies, respectively.

An auto-encoder (AE) was used [36], which is a type of ANN used to inexpensively grasp data. By training the network, an AE attempts to discover a representation for a dataset to disregard “noise” signals to limit the number of parameters. The encoder, message, and decoder are the three components that make up an auto-encoder. In cybersecurity, a deep AE can be utilized to develop a viable security model. Therefore, the AE-based feature learning (FL) model surpasses other sophisticated algorithms. Compared to other complex algorithms, the AE-based FL model employs the lowest security measures. The approach is more productive and practical, particularly in small areas such as the Internet of Things, due to the dense and latent representation of security characteristics [37]. The authors of [38] demonstrated the efficacy of an AE-based FL prototype for malware categorization and detection. A Deep-AE-based anomaly detection model was proposed by the authors in [39] to develop an efficient ID model using the Restricted Boltzmann Machine (RBM).

On the UNSW-NB15 dataset, Zakir et al. [40] investigated the performance of four common classifiers for binary classification: Support Vector Machine (SVM), Random Forest, Naive Bayes, and Decision Tree. The researchers utilized One-hot encoding to convert categorical data to attribute values and then performed machine learning on the

complete feature set. The results of the experiments indicated that the researchers attained accuracies of 79.59%, 66%, 76%, and 78% on SVM, Naive Bayes, Random Forest, and Decision Tree, respectively.

To address the limitations in the development of IDS feature selection, hybrid approaches have emerged. These strategies combine the filtering and wrapping processes to make use of and increase the effectiveness of both techniques, as well as to improve predictions with enhanced computation. Thus, Song et al. [41] presented a model that combines chi-square with RF to build an intrusion detection hybrid feature selection (FS) approach. Furthermore, Wang et al. [42] considered two schemes: one with a Naive Bayes classifier integrated with information acquisition and the other with decision-making. A hybrid strategy integrating the linear correlation analysis approach with the cuttlefish algorithm was recently integrated with a decision tree as a classifier [42]. The fundamental disadvantage of this class of techniques is that the wrapping method is dependent on the performance of the filter method, which is combined with the hybrid technique. To put it another way, the wrapper methodology can only operate with the components delivered by the filter function. In this instance, there is a possibility that informative characteristics will be filtered away and will not be included for wrapper evaluation.

Zhang et al. [43] introduced a neural-network-based anomaly detection model on the LeNet 5 convolutional neural network (CNN) and the Long short-term memory (LSTM) feature reduction algorithm. CICIDS 2017 and CTU datasets were used in the experiments, which used binary and multi-classification. The use of CNN, LSTM, and hybrid combinations resulted in increased efficiency on both binary and multi-classification examinations.

Furthermore, Aydin et al. [44] presented an approach where they integrated the two following methodologies: (1) Anomaly Detection (PHAD) in a packet header; (2) Network Anomaly Detection (NETAD) on a network using the IDS Snort on a signature basis. Both approaches are anomaly-based intrusion detection systems. The proposed hybrid IDS was assessed using IDEVAL data, which indicated that the number of attacks detected increased significantly in comparison to signature-based systems.

The literature demonstrates that malicious threats arise and evolve frequently; hence, the network requires a very effective security solution. Due to the complexity of new attacks, the present models are incapable of detecting them. Deep learning has enabled researchers to investigate new fields of inquiry. Deep-Learning-based techniques necessitate little input while exploring every possible set of features. Using this technique in intrusion detection can assist in the detection of such malicious attacks. The main aim of the paper is to accurately detect network intrusions by employing a model that can identify malicious traffic using deep learning and detect different types of intrusion attacks, as a deep-learning-based intelligent approach is proposed in this research. The proposed model overcomes the issues in existing models and provides promising results.

4. Proposed HDLNIDS Model

Deep learning for the detection of malicious traffic enables the detection of various changes in traffic, which in turn enhances the performance and allows only normal traffic to pass into the system.

Deep learning can assist in the identification of hostile intrusions for the detection of rising attacks. We proposed a model based on deep learning for an intrusion detection system that is illustrated in Figure 1. It comprises two learning phases, which are represented below. We proposed this method to build an IDS utilizing an HDLNIDS-based deep learning technique. Our suggested HDLNIDS is better in terms of computation while employing full-featured datasets and provides enhanced accuracy with a low likelihood of failure.

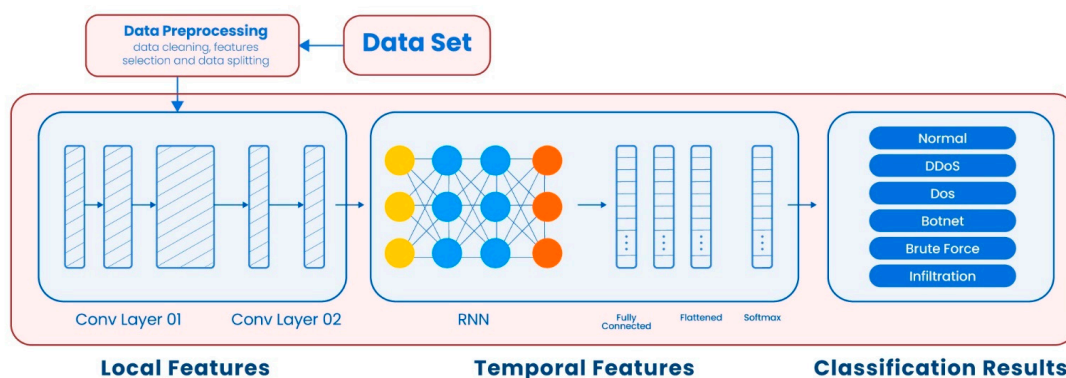


Figure 1. Proposed HDLNIDS model.

With a massive data processing architecture, HDLNIDS learning concentrates on tackling real-world ID challenges. Solving such a challenge is difficult due to a lack of time and space. Big data now has massive and expanding quantities; however, it requires enormous amounts of energy, resources, and a computing device to help with the training process that can deal with significant data correctly.

Combining the RNN with a CNN-DL model, HDLNIDS reduces the aforementioned issues. Figure 1 depicts the HDLNIDS in further detail. According to the HDLNIDS overview, a CNN comprises two basic components: a feature extractor comes first, followed by a classifier. The feature extractor comprises two layers: a convolution comprising of the twin set of layers and a pooling layer. The obtained output called the feature map is fed into the second component for classification. CNN adopts the local characteristics in this way. The drawback is that it overlooks the time dependency of essential features. As a result, we added recurrent layers following the CNN layers to capture both spatial and temporal data more adequately. These four layers of RNN were added to make it more efficient in terms of accuracy and computation. Using this, we were able to effectively manage the disappearance and inflating gradient issues, which increases the capacity to capture temporal and spatial correlations and efficient learning from varied extent patterns.

The CNN first processes the input in the HDLNIDS network, and the output is subsequently sent to the recurrent layers, which produce sequences at each timestep, allowing for the modeling of both spatial and temporal characteristics. The resultant sequential vector is then input into a layer of SoftMax for the probability distribution over the categories after passing through a set of two fully linked layers.

4.1. Data Preprocessing

In the data pre-processing phase, initially, the network traffic was categorized and preprocessed. All necessary conversions for the HDLNIDS and IDS-compatible file formats were performed during pre-processing. The different characteristics, such as timestamps and network Internet Protocol (IP) addresses, have little effect on whether network traffic is malicious in the original CICIDS-2018. As timestamp features are utilized to keep track of when malicious communication occurs and give only minor assistance while training the algorithm, we eliminated them during the pre-processing step. Similar to an anomalous intrusion detection system (AIDS), the majority of traffic should be classed based on its behavior, without bias or competing with the IP address; thus, we deleted the IP address features as well. Pandas, NumPy, and Scikit-learn packages within the Python language were used to implement data pre-processing tasks.

The subject of class inequality has received significant attention from the research community. A class imbalance is caused by inadequate data distribution; one class has the majority of the samples, while others have comparably few. Because of unlimited data values and imbalanced classes, the classification issue becomes more challenging as data dimensionality grows. Bedi et al. [45] used a variety of ML methods to address the class imbalance problem. Thabtah et al. [46] investigated numerous methods for the

problem of class imbalance. Most algorithms target the majority of data samples while missing the minority of data samples. Hence, minority samples emerge in an irregular but consistent manner. The basic strategies to resolve the data problem include data preprocessing and feature selection, and each approach has advantages and disadvantages. The ID dataset has a problem with high-dimensional imbalance, which includes lacking interesting features, attribute values, and the only availability of cumulative data. The data appear to be unreliable, with inaccuracies and anomalies, as well as unpredictable, with variations in codes or names. To overcome the imbalance problem, we utilized over-sampling, which included increasing the frequency of occurrences among the minority group by indiscriminately reproducing them to enhance the representation of the minority group in the dataset.

Although there is a risk of overfitting with this process, no records were lost, and the over-sampling approach outperformed the under-sampling option. Table 1 presents the extracted features during preprocessing phase.

Table 1. An overview of the CICIDS-2018 ID dataset's extracted characteristics.

Features	Description
Fw-iat-min	The shortest delay between two packets supplied in an onward route.
Bw-iat-tot	The total lag time between two packets transmitted via a back channel.
Bw-iat-avg	The average time between two packets transmitted through a back channel.
Bw-iat-std	The average time between two packets transmitted in the reverse direction.
Bw-iat-max	The highest time between two packets transmitted in the reverse direction.
Bw-iat-min	The shortest time between two packets transmitted in a forward direction.
Bw-iat-min	The lowest time between two packets transmitted in a reverse direction.
Fw-pkt-l-avg	The average quantity of data in a packet in an upward direction
Fw-pkt-l-min	The lowest volume of the package
Tot-bw-pk	Overall data packets in a back channel
Tot-l-fw-pkt	Overall size of the packet in an up channel
Tot-fw-pk	Forwardly aggregate data packets
Fl-iat-max	The highest-duration period between two flows
Fl-dur	Interval of flow

4.2. Model Training and Testing

Model training is an important step after data preprocessing to extract meaningful features from the training set. For this purpose, we divided the dataset into 80, 10, and 10 training, testing, and validation sets, respectively. To train an efficient model, we ensured that all of the class samples were present in the training, validation, and testing set. During the training process, the model was tested on the validation set throughout the process to assess the performance of the model, and this process helps to obtain better results when unseen data are provided to the model.

5. Dataset Detail

Selecting appropriate ID data to analyze the ID system is crucial, so we chose the data preceding the simulation of the suggested methodology. Even though many ID databases are openly available, several of them contain outdated, inaccurate, and unreproducible intrusions. To address these shortcomings and generate current traffic patterns, the Amazon Web services (AWS) platform created the well-known CICIDS-2018 [47] dataset. The CICIDS 2018 intrusion dataset depicts real-time network behavior and includes a variety of intrusion modes. Furthermore, it is spread as a full network that encompasses all of the internal networks traced to compute data packet payloads. These qualities of the CICIDS-2018 dataset compelled us to use it in our research for the proposed system.

This dataset provides many intrusion patterns that may be applied to a variety of network protocols and topologies about safety and security. This dataset is an upgrade to the CICIDS-2017 dataset.

CICIDS-2018 is a publicly accessible dataset with two patterns and seven intrusion techniques at the moment. Multiple data states were gathered, and raw data were refreshed. CICIDS-2018 contains 80 statistical features measured in forward and reverse modes, such as volume, packet length, and the number of bytes. Finally, the dataset, which had around 5 million entries, was made available to all researchers over the internet. The CICIDS-2018 dataset is available in PCAP and CSV formats. In this research, we considered the use of CSV format, whereas the PCAP format is utilized to extract innovative features [48,49]. This CICIDS-2018 dataset includes various categories of attacks:

- Brute-force DOS attacks;
- Botnet;
- Heartbleed;
- DDOS attacks;
- Brute-force SSH;
- Infiltration;
- Web attacks.

The dataset framework comprises 50 systems, whereas the attacking firms comprise 31 servers and 421 endpoints. CICIDS-2018 data provide AWS-recorded network traffic and a system log containing 80 retrieved parameters using CICFlowMeter-V3. The CICIDS-2018 dataset is approximately 400 GB in size, which is greater than the CICIDS-2017 dataset. Table 1 shows a few retrieved characteristics from the CICIDS-2018 dataset. The sample size of the CICIDS-2018 dataset was compared to that of CICIDS-2017. Table 2 displays the outcomes of both datasets, notably in the Botnet and Infiltration assaults, where it increased by 143 and 4497, respectively. Moreover, the number of Internet Attacks provided in CICIDS-2018 is quite low (928). Table 2 presents the comparison between CICIDS-2018 and CICIDS-2017 datasets.

Table 2. Comparison of CICIDS-2018 ID dataset with CICIDS-2017.

Dataset	Normal	DDoS	Brute Force	Infiltration	Dos	Port Scan	Web Attacks	Botnet
CICIDS-2017	1,743,179	128,027	13,835	36	252,661	158,930	2180	1966
CICIDS-2018	6,112,151	687,742	380,949	161,934	654,301	-	928	286,191

6. Performance Analysis

In this subsection, we explain the simulation environment used to test our proposed model along with performance parameters and results.

6.1. Simulation Setup

To validate the efficacy of the suggested ID strategy, we implemented the proposed technique in Python using deep learning. The experiment was carried out on a computer (Core i7, 64-bit, 24 GB RAM, 64-core CPU). To improve pipeline speed, the model was trained using an NVIDIA GTX 2080 Ti GPU. To evaluate the proposed models, the dataset was divided into training and testing datasets separately. We utilized the training dataset to train the proposed model to make it effective. Then, the testing set was used to assess the efficiency of the proposed model. We utilized the CICIDS-2018 dataset to demonstrate the effectiveness of the proposed solutions. The network traffic included both malicious and normal information, which the proposed model categorized into malicious and non-malicious categories, respectively. To obtain high accuracy and a low FAR value, the proposed technique decreases computational complexity by utilizing rich characteristics from the CICIDS-2018 dataset. Even though the CSE-CIC-IDS2018 data were utilized for training and testing, the model was tested with 10-fold cross-validation in each step, and each model was trained on the lot size using first-order gradient-based optimization algorithms such as RMSprop and Ada Max with different learning rates, while various combinations of hyperparameters were used to optimize the actual network packet using a search algorithm and 10-fold cross-validation. We included Gaussian noise layers after

convolutional and recurrent layers to improve model flexibility in terms of performance and to prevent overfitting. Table 3 presents the hardware specifications used for this research.

Table 3. Hardware Specification.

Machine	Specification
Processor	Core i7
Type	64-Bit
RAM	64-GB
Core	64-core
Type	CPU
Graphic Card	NVIDIA GTX 1080 Ti

6.2. Evaluation Metrics

A confusion matrix (CM) assists in determining the actual and expected classification. The classification result is divided into two categories: normal and abnormal. Four crucial states in the confusion matrix must be measured.

- True Positive (TP): this implies demonstrating that the model is accurate and representative and that it predicts favorable results.
- False negative (FN): this is defined as an inaccurate prediction. It accurately classifies malicious situations as normal, whereas it predicts bad outcomes wrongly.
- False positive (FP): when the number of attacks seen is typical, the model predicts a favorable result.
- True negative (TN): this refers to events that are appropriately identified as an assault and forecasts unfavorable outcomes.

The following performance matrices were used to evaluate the performance of the proposed system in terms of accuracy, precision, recall, and F1 score as mentioned below in Equations (1)–(4), respectively:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{Total Samples}} \quad (1)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

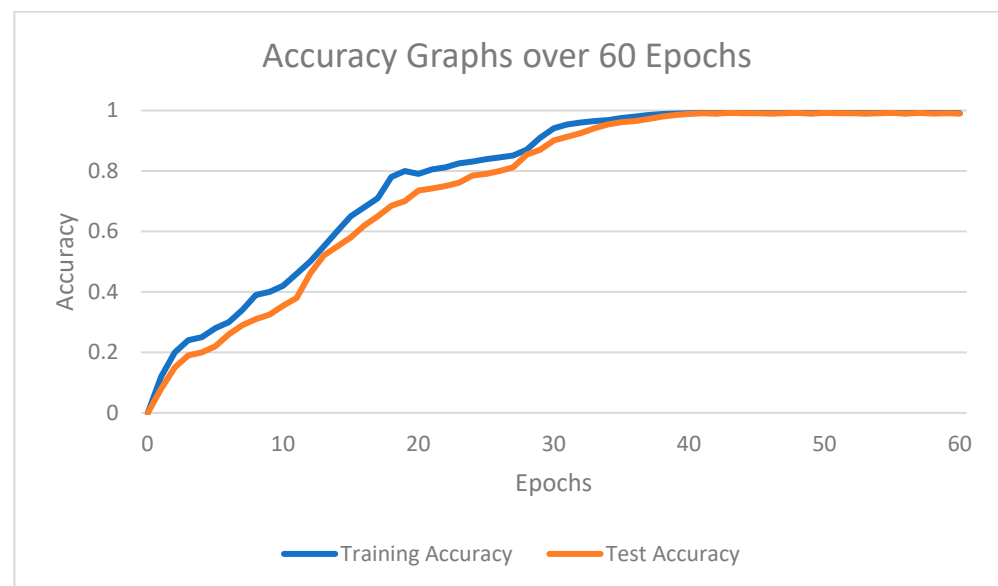
6.3. Results and Evaluation

Experiments were conducted to evaluate the effectiveness of HDLNIDS in terms of accuracy, precision, recall, F1 measure, and the loss graph. We used 10-fold cross-validation to evaluate the ability of the model on new data as shown in Table 4.

Table 4. Evaluation and Characteristics of Architecture on each Fold.

Folds	Precision (%)	Recall (%)	F-Measure (%)	Accuracy (%)
Fold 1	98.33	98.91	98.81	98.95
Fold 2	99.14	98.82	99.63	98.76
Fold 3	99.45	99.55	98.64	98.56
Fold 4	99.05	99.43	99.85	99.02
Fold 5	98.85	99.16	99.12	99.23
Fold 6	98.77	99.04	98.83	98.84
Fold 7	98.83	98.92	98.87	98.67
Fold 8	98.79	99.32	99.15	98.9
Fold 9	98.82	99.18	98.71	99.18
Fold 10	98.60	99.16	98.83	98.91
Average	98.63	99.14	99.03	98.90

Figure 2 displays the results in terms of accuracy, which indicate that the proposed model's accuracy continues increasing with time. Initially, the accuracy of the model is observed as low, but it continues increasing over epochs. This is due to the learning ability of the proposed model that makes it stable and more competent to classify normal and malicious traffic.

**Figure 2.** Accuracy of proposed deep learning model with respect to epochs.

The results of the proposed model for the loss graph are shown in Figure 3. This indicates that the proposed model loss graph continues decreasing with time. Initially, the model's loss graph is observed to be high, but it steadily decreases over time. This is due to the proposed model's ability to learn, which makes it more stable and competent at classifying normal and malicious traffic.

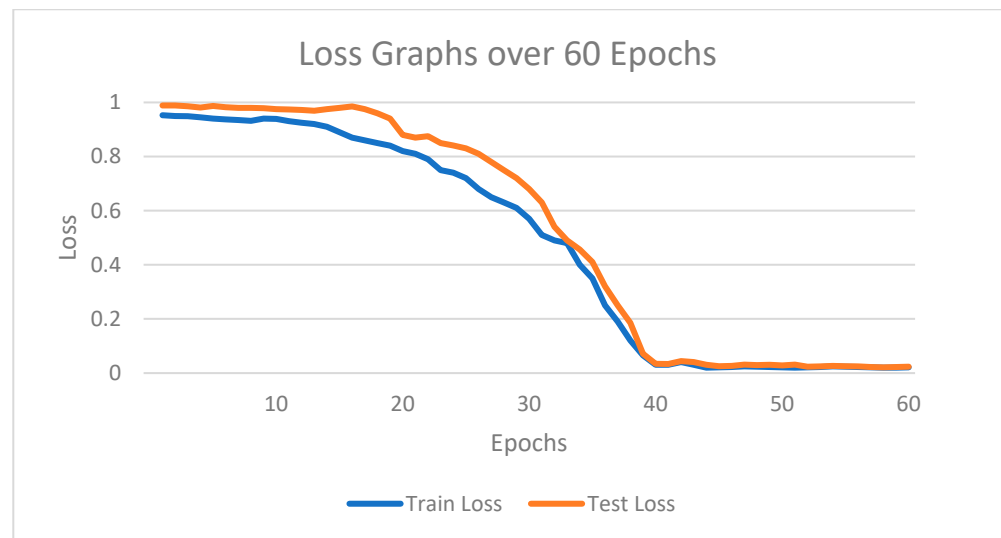


Figure 3. Loss graph of proposed deep learning model with respect to epochs.

6.4. Characteristics of Each Fold

We present the performance of fold-wise computation in Table 4, which describes the results in terms of precision, recall, F-measure, and accuracy achieved against each fold. With each fold, the results continue improving, and the model achieves an average of 98.63% precision, 99.14% recall, 99.03% F-measure, and 98.90% accuracy.

7. Performance Analysis

In this section, we compare the proposed technique with existing techniques in terms of the dataset, and the methodology they used to detect the network intrusion. We also reviewed the technique to detect the limitations of the literature. A detailed comparison with existing techniques is presented in Table 5. Most of the authors used the NSL-KDD, KDD99, UNSW-NB15, and AWIS dataset, while the approaches used were the Sparse autoencoder, conjugate gradient algorithm (CGA), deep belief networks, DNN, CNN, RNN, and LSTM [48–54]. Most of the research conducted used limited data, while, in [49], the performance could be improved by improving the sampling methodology. There are different types of networks attacks, and no study caters to all of them; most research focuses on the detection of binary classification while some research considers multiclass classification within a limited scope. From Table 5, it is evident that our proposed approach achieves significant results as compared to existing techniques.

Table 5. Comparison with Existing Techniques.

Author	Dataset	Methodology	Results	Limitations
Javaid et al. [48]	NSL-KDD	Sparse Taught Learning with Sparse autoencoder	Binary Classification, 85.44% Precision, 95.95% Recall, 90.4% F-measure, 88.39% Accuracy	Implementation of an efficient NIDS is required to handle a multi-class problem
Wijesty et al. [49]	KDD-Cup1999	Conjugate Gradient algorithm (CGA)	Reported an accuracy of 93.2% and 54.13% for binary and multi-class classification, respectively	The use of the sampling method can improve the performance of the proposed technique.
Shone et al. [50]	KDD99, NSL-KDD	RF Classification, Nonsymmetric deep autoencoder (NDAE), DL Stacked NDAEs	Accuracy of 89.22%, Precision of 92.97% Recall of 89.22%, and F-Score of 90.76%.	This method does not apply to handling zero-day attacks.
Caminero et al. [51]	AWID, NSL-KDD	Adversarial environment reinforcement learning	Reported accuracy of 80.16%, precision of 79.74%, Recall of 80.16%, F-Score of 79.40%	-

Table 5. Cont.

Author	Dataset	Methodology	Results	Limitations
Feng et al. [52]	KDD99	DNN, LSTM, CNN	Reported accuracy of 98.5%, Precision of 97.63%, and recall of 99.59% (multi-class classification)	Limited to 3 classes only: SQL, XSS, and DoS
Yang et al. [53]	UNSW-NB15, NSL-KDD	Deep Belief Networks (DBF) and modified density peak clustering algorithm	Reported FPR of 2.62% and accuracy of 82.08% (multi-class classification)	They synthesized U2R and R2L attacks to increase the model performance.
Aminanto et al. [54]	AWID	Sparse Autoencoder	Reported F1-score of 89.06%, detection Rate of 92.18%, and accuracy of 94.81% for multi-class classification	-
Kshirsagar et al. [55]	CICIDS 2018	Rule-Based Classifiers	Reported accuracy 99.9%	Lack of detail about experiments and unknown measure of buildup time.
Bharati et al. [56]	CICIDS 2018	Random Forest	Reported accuracy 99.9%	Classification information is not available.
Alani et al. [57]	UNSW-NB15	Various ML algorithms	Reported average accuracy of 99%	Conducted experiments using hand-engineered methods.
Proposed	CICIDS-2018	CNN, RNN	Average Accuracy of 98.90%, F-measure of 99.03, precision of 98.64%, recall of 99.15%	-

By performing extensive experiments, we determined the optimal parameters for our proposed model: it took 3.06 s per epoch for training and validation. The proposed model was trained for 60 epochs due to early stopping and it took approximately 3 min in total to train the model.

8. Conclusions and Future Work

Currently, the most urgent issue facing modern society is network attacks. All networks are susceptible to network risks, regardless of size. A network must have an intrusion detection system (ID) for detecting and mitigating hostile attacks. As malicious threats continually emerge and evolve, the network requires a highly advanced security solution.

Using deep learning to detect malicious traffic enables the detection of various changes in traffic, which in turn enhances the performance and allows only normal traffic to pass into the system.

Due to this, the development of an effective and intelligent ID system is of the utmost importance. In this study, a hybrid ID framework based on Deep Learning is created using a convolutional recurrent neural network (CRNN) that detects hostile network attacks. The model is built by combining an RNN with a CNN in which two convolutional layers are followed up by various RNN layers; the result is then passed into fully connected, flattened, and SoftMax layers that make the model capable of detecting and classifying traffic. To enhance the accuracy and predictability of the ID system, the CNN collects local features via convolution, whereas a deep-layered RNN captures HDLNIDS features. Experiments are conducted using publicly available intrusion detection data, in particular, the modern and realistic CICIDS-2018 data, to determine the efficacy of the proposed HDLNIDS system. The simulation results demonstrate that the proposed HDLNIDS gives promising results in terms of accuracy and data loss.

In the near future, we want to expand our model with more parameters that contribute to enhanced performance and detection, by designing more effective algorithms for other malicious network traffic using multiple deep learning techniques. We will also analyze and expand our model's capacity to handle zero-day assaults as our initial avenue for improvement. Furthermore, we will seek to expand upon our current assessments by employing actual backbone network traffic to illustrate the expanded model's value.

Author Contributions: Conceptualization, E.U.H.Q. and T.Z.; data curation, E.U.H.Q., M.H.F. and T.Z.; formal analysis, E.U.H.Q., M.H.F. and T.Z.; funding acquisition, T.Z.; methodology, E.U.H.Q.; project administration, T.Z.; resources, T.Z.; software, E.U.H.Q. and M.H.F.; supervision, T.Z.; validation, E.U.H.Q., M.H.F. and T.Z.; visualization, E.U.H.Q. and M.H.F.; writing—original draft, E.U.H.Q., M.H.F. and T.Z.; writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Security Research Center at Naif Arab University for Security Sciences (Project No. SRC-PR2-02).

Data Availability Statement: The data presented in this study are openly available in <https://www.unb.ca/cic/datasets/ids-2018.html> (accessed on 16 January 2023) at [47].

Acknowledgments: The author would like to express their deep thanks to the Vice Presidency for Scientific Research at Naif Arab University for Security Sciences for their kind encouragement of this work.

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

References

1. Anderson, J.P. Technical Report. In *Computer Security Threat Monitoring and Surveillance*; James, P., Ed.; Anderson Company: Washington, DC, USA, 1980.
2. Liao, Y.; Vemuri, V.R. Use of k-nearest neighbor classifier for intrusion detection. *Comput. Secur.* **2002**, *21*, 439–448. [CrossRef]
3. Kim, G.; Lee, S.; Kim, S. A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Syst. Appl.* **2013**, *41*, 1690–1700. [CrossRef]
4. De la Hoz, E.; De La Hoz, E.; Ortiz, A.; Ortega, J.; Prieto, B. PCA_ltering and probabilistic SOM for network intrusion detection. *Neuro-Computing* **2015**, *164*, 71–81.
5. Sen, R.; Chattopadhyay, M.; Sen, N. An efficient approach to develop an intrusion detection system based on multi layer backpropagation neural network algorithm: IDS using BPNN algorithm. In Proceedings of the 2015 ACM SIGMIS Conference on Computers and People Research, Newport Beach, CA, USA, 4–6 June 2015; pp. 105–108.
6. Koc, L.; Mazzuchi, T.A.; Sarkani, S. A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier. *Expert Syst. Appl.* **2012**, *39*, 13492–13500. [CrossRef]
7. Khan, M.A.; Kim, Y. Deep Learning-Based Hybrid Intelligent Intrusion Detection System. *Comput. Mater. Contin.* **2021**, *68*, 671–687.
8. Devi, B.T.; Thirumaleshwari, S.S.; Jabbar, M.A. An Appraisal over Intrusion Detection Systems in Cloud Computing Security Attacks. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; pp. 722–727.
9. Thaseen, I.S.; Poorva, B.; Ushasree, P.S. Network Intrusion Detection using Machine Learning Techniques. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Tamil Nadu, India, 24–25 February 2020; pp. 1–7.
10. Yin, C.; Zhu, Y.; Fei, J.; He, X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
11. Soheil-Khah, S.; Marteau, P.-F.; Bechet, N. Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset. In Proceedings of the 2018 1st International Conference on Data Intelligence and Security (ICDIS), South Padre Island, TX, USA, 8–10 April 2018; pp. 219–226.
12. Folino, F.; Folino, G.; Guarascio, M.; Pisani, F.; Pontieri, L. On learning effective ensembles of deep neural networks for intrusion detection. *Inf. Fusion* **2021**, *72*, 48–69. [CrossRef]
13. Tama, B.A.; Lim, S. Ensemble learning for intrusion detection systems: A systematic mapping study and cross-benchmark evaluation. *Comput. Sci. Rev.* **2021**, *39*, 100357. [CrossRef]
14. Kim, K.; Aminanto, M.E.; Tanuwidjaja, H.C. *Network Intrusion Detection Using Deep Learning: A Feature Learning Approach*; Springer: Berlin/Heidelberg, Germany, 2018.

15. Avci, O.; Abdeljaber, O.; Kiranyaz, S.; Hussein, M.; Gabbouj, M.; Inman, D.J. A review of vibration-based damage detection in civil structures: From traditional methods to Machine Learning and Deep Learning applications. *Mech. Syst. Signal Process.* **2021**, *147*, 107077. [\[CrossRef\]](#)
16. Kumar, K.P.M.; Saravanan, M.; Thenmozhi, M.; Vijayakumar, K. Intrusion detection system based on GA-fuzzy classifier for detecting malicious attacks. *Concurr. Comput. Pr. Exp.* **2021**, *33*, 5242.
17. Khan, M. HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. *Processes* **2021**, *9*, 834. [\[CrossRef\]](#)
18. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [\[CrossRef\]](#)
19. Siddiqui, M.K.; Naahid, S. Analysis of KDD CUP 99 Dataset using Clustering based Data Mining. *Int. J. Database Theory Appl.* **2013**, *6*, 23–34. [\[CrossRef\]](#)
20. Binbusayyis, A.; Vaiyapuri, T. Identifying and Benchmarking Key Features for Cyber Intrusion Detection: An Ensemble Approach. *IEEE Access* **2019**, *7*, 106495–106513. [\[CrossRef\]](#)
21. Bhavani, T.T.; Rao, M.K.; Reddy, A.M. Network Intrusion Detection System Using Random Forest and Decision Tree Machine Learning Techniques. In Proceedings of the Distributed Computing and Artificial Intelligence, 13th International Conference, Sevilla, Spain, 1–3 June 2016; Springer: Berlin/Heidelberg, Germany, 2019; pp. 637–643.
22. Karatas, G.; Demir, O.; Sahingoz, O.K. Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. *IEEE Access* **2020**, *8*, 32150–32162. [\[CrossRef\]](#)
23. Xu, H.; Przystupa, K.; Fang, C.; Marciniak, A.; Kochan, O.; Beshley, M. A Combination Strategy of Feature Selection Based on an Integrated Optimization Algorithm and Weighted K-Nearest Neighbor to Improve the Performance of Network Intrusion Detection. *Electronics* **2020**, *9*, 1206. [\[CrossRef\]](#)
24. Bhati, B.S.; Rai, C.S. Analysis of Support Vector Machine-based Intrusion Detection Techniques. *Arab. J. Sci. Eng.* **2019**, *45*, 2371–2383. [\[CrossRef\]](#)
25. Thaseen, I.S.; Banu, J.S.; Lavanya, K.; Ghalib, M.R.; Abhishek, K. An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, 4014.
26. Waskle, S.; Parashar, L.; Singh, U. Intrusion Detection System Using PCA with Random Forest Approach. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2–4 July 2020; pp. 803–808.
27. Alqahtani, H.; Sarker, I.H.; Kalim, A.; Hossain, S.M.M.; Ikhlaq, S.; Hossain, S. Cyber Intrusion Detection Using Machine Learning Classification Techniques. In *Communications in Computer and Information Science*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2020; Volume 1235, pp. 121–131.
28. Qazi, E.U.H.; Imran, M.; Haider, N.; Shoaib, M.; Razzak, I. An intelligent and efficient network intrusion detection system using deep learning. *Comput. Electr. Eng.* **2022**, *99*, 107764. [\[CrossRef\]](#)
29. Qazi, E.U.H.; Almorjan, A.; Zia, T. A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection. *Appl. Sci.* **2022**, *12*, 7986. [\[CrossRef\]](#)
30. Ahmad, I.; Ul Haq, Q.E.; Imran, M.; Alassafi, M.O.; AlGhamdi, R.A. An Efficient Network Intrusion Detection and Classification System. *Mathematics* **2022**, *10*, 530. [\[CrossRef\]](#)
31. Girdler, T.; Vassilakis, V.G. Implementing an intrusion detection and prevention system using Software-Defined Networking: Defending against ARP spoofing attacks and Blacklisted MAC Addresses. *Comput. Electr. Eng.* **2021**, *90*, 106990. [\[CrossRef\]](#)
32. Idhammad, M.; Karim, A.; Belouch, M. Distributed intrusion detection system for cloud environments based on data mining techniques. *Procedia Comput. Sci.* **2018**, *127*, 35–41. [\[CrossRef\]](#)
33. Imran, R.; Jamil, F.; Kim, D. An Ensemble of Prediction and Learning Mechanism for Improving Accuracy of Anomaly Detection in Network Intrusion Environments. *Sustainability* **2021**, *13*, 10057. [\[CrossRef\]](#)
34. Biney, G.; Okyere, G.A.; Alhassan, A. Adaptive scheme for ANOVA models. *J. Adv. Math. Comput. Sci.* **2020**, *35*, 12–23. [\[CrossRef\]](#)
35. Feir-Walsh, B.J.; Toothaker, L.E. An empirical comparison of the ANOVA F-test, normal scores test and Kruskal–Wallis test under violation of assumptions. *Educ. Psychol. Meas.* **1974**, *34*, 789–799. [\[CrossRef\]](#)
36. Guijuan, Z.; Yang, L.; Xiaoning, J. A survey of autoencoder-based recommender systems. *Front. Comput. Sci.* **2020**, *14*, 430–450.
37. Liu, J.; Song, K.; Feng, M.; Yan, Y.; Tu, Z.; Zhu, L. Semi-supervised anomaly detection with dual prototypes autoencoder for industrial surface inspection. *Opt. Lasers Eng.* **2021**, *136*, 106324. [\[CrossRef\]](#)
38. Yousefi-Azar, M.; Varadharajan, V.; Hamey, L.; Tupakula, U. Autoencoder-based feature learning for cybersecurity applications. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 3854–3861.
39. Khan, M.A.; Kim, J. Toward Developing Efficient Conv-AE-Based Intrusion Detection System Using Heterogeneous Dataset. *Electronics* **2020**, *9*, 1771. [\[CrossRef\]](#)
40. Hossain, Z.; Sourov, M.M.R.; Khan, M.; Rahman, P. Network Intrusion Detection using Machine Learning Approaches. In Proceedings of the 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 11–13 November 2021.
41. Song, J.; Zhao, W.; Liu, Q.; Wang, X. Hybrid feature selection for supporting lightweight intrusion detection systems. *J. Physics: Conf. Ser.* **2017**, *887*, 012031. [\[CrossRef\]](#)

42. Wang, W.; He, Y.; Liu, J.; Gombault, S. Constructing important features from massive network traffic for lightweight intrusion detection. *IET Inf. Secur.* **2015**, *9*, 374–379. [[CrossRef](#)]
43. Zhang, Y.; Chen, X.; Jin, L.; Wang, X.; Guo, D. Network Intrusion Detection: Based on Deep Hierarchical Network and Original Flow Data. *IEEE Access* **2019**, *7*, 37004–37016. [[CrossRef](#)]
44. Aydin, M.; Zaim, A.H.; Ceylan, K.G. A hybrid intrusion detection system design for computer network security. *Comput. Electr. Eng.* **2009**, *35*, 517–526. [[CrossRef](#)]
45. Bedi, P.; Gupta, N.; Jindal, V. I-SiamIDS: An improved Siam-IDS for handling class imbalance in network-based intrusion detection systems. *Appl. Intell.* **2021**, *51*, 1133–1151. [[CrossRef](#)]
46. Thabtah, F.; Hammoud, S.; Kamalov, F.; Gonsalves, A. Data imbalance in classification: Experimental evaluation. *Inf. Sci.* **2020**, *513*, 429–441. [[CrossRef](#)]
47. A Collaborative Project between the Communications Security Establishment (CSE) & the Canadian Institute for Cybersecurity (CIC). Available online: <https://www.unb.ca/cic/datasets/ids-2018.html> (accessed on 31 March 2021).
48. Javaid, A.; Niyaz, Q.; Sun, W.; Alam, M. A deep learning approach for network intrusion detection system. In Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies (Formerly BIONETICS), New York, NY, USA, 24 May 2016; pp. 21–26.
49. Wisesty, U.N. Comparative study of conjugate gradient to optimize the learning process of neural network for intrusion detection system (ids). In Proceedings of the 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, 25–26 October 2017; pp. 459–464.
50. Shone, N.; Ngoc, T.N.; Phai, V.D.; Shi, Q. A deep learning approach to network intrusion detection. *IEEE Trans. Emerg. Top. Comput. Intell.* **2018**, *2*, 41–50. [[CrossRef](#)]
51. Caminero, G.; Lopez-Martin, M.; Carro, B. Adversarial environment reinforcement learning algorithm for intrusion detection. *Comput. Netw.* **2019**, *159*, 96–109. [[CrossRef](#)]
52. Feng, F.; Liu, X.; Yong, B.; Zhou, R.; Zhou, Q. Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device. *Ad. Hoc. Netw.* **2019**, *84*, 82–89. [[CrossRef](#)]
53. Yang, S.; Li, M.; Liu, X.; Zheng, J. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 721–736. [[CrossRef](#)]
54. Aminanto, M.E.; Kim, K. Improving detection of wi-fi impersonation by fully unsupervised deep learning. In *International Workshop on Information Security Applications*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 212–223.
55. Kshirsagar, D.; Shaikh, J.M. Intrusion Detection Using Rule-Based Machine Learning Algorithms. In Proceedings of the 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 19–21 September 2019; pp. 1–4. [[CrossRef](#)]
56. Bharati, M.P.; Tamane, S. NIDS-Network Intrusion Detection System Based on Deep and Machine Learning Frameworks with CICIDS 2018 using Cloud Computing. In Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 30–31 October 2020; pp. 27–30. [[CrossRef](#)]
57. Alani, M.M. Implementation-Oriented Feature Selection in UNSW-NB15 Intrusion Detection Dataset. In *Intelligent Systems Design and Applications*; Abraham, A., Gandhi, N., Hanne, T., Hong, T.P., Nogueira Rios, T., Ding, W., Eds.; ISDA 2021. Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2022; Volume 418. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.