International Journal of Safety and Security Engineering
Vol. 13, No. 4, August, 2023, pp. 587-595
Journal homepage: http://iieta.org/journals/ijsse

Development of a Malicious Network Traffic Intrusion Detection System Using Deep Learning

Olisaemeka F. Isife[1], Kennedy Okokpujie[1,2*], Imhade P. Okokpujie[3,4], Roselyn E. Subair[5], Akingunsoye Adenugba Vincent[6], Morayo E. Awomoyi[7]

[1] Department of Electrical and Information Engineering, Covenant University, Ota 112101, Nigeria
[2] Africa Centre of Excellence for Innovative & Transformative STEM Education, Lagos State University, Ojo 102101, Nigeria
[3] Department of Mechanical and Mechatronics Engineering, Afe Babalola University, Ado Ekiti 360001, Nigeria
[4] Department of Mechanical and Industrial Engineering Technology, University of Johannesburg, Johannesburg 2028, South Africa
[5] University Librarian, Afe Babalola University, Ado-Ekiti 360001, Nigeria
[6] OVA Foundation, Millington 21651, USA
[7] US School of International Service, American University, Washington 20016, USA

Corresponding Author Email: kennedy.okokpujie@covenantuniversity.edu.ng

https://doi.org/10.18280/ijsse.130401

Received: 26 February 2023
Revised: 14 July 2023
Accepted: 2 August 2023
Available online: 28 September 2023

Keywords:
cybersecurity, intrusion detection, intrusion detection system, deep learning, CICIDS 2017 dataset, CIDDS dataset

ABSTRACT

With the exponential surge in the number of internet-connected devices, the attack surface for potential cyber threats has correspondingly expanded. Such a landscape necessitates the evolution of intrusion detection systems to counter the increasingly sophisticated mechanisms employed by cyber attackers. Traditional machine learning methods, coupled with existing deep learning implementations, are observed to exhibit limited proficiency due to their reliance on outdated datasets. Their performance is further compromised by elevated false positive rates, decreased detection rates, and an inability to efficiently detect novel attacks. In an attempt to address these challenges, this study proposes a deep learning-based system specifically designed for the detection of malicious network traffic. Three distinct deep learning models were employed: Deep Neural Networks (DNN), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU). These models were trained using two contemporary benchmark intrusion detection datasets: the CICIDS 2017 and the Coburg Intrusion Detection Data Sets (CIDDS). A robust preprocessing procedure was conducted to merge these datasets based on common and essential features, creating a comprehensive dataset for model training. Two separate experimental setups were utilized to configure these models. Among the three models, the LSTM displayed superior performance in both experimental configurations. It achieved an accuracy of 98.09%, a precision of 98.14%, an F1-Score of 98.09%, a True Positive Rate (TPR) of 98.05%, a True Negative Rate (TNR) of 99.69%, a False Positive Rate (FPR) of 0.31%, and a False Negative Rate (FNR) of 1.95%.

1. INTRODUCTION

The pervasive incorporation of Digital Communication Technology (DCT) into human society and organizational structures, both in the private and governmental sectors, has become a fundamental driver of economic growth and infrastructure development [1]. The global spread of internet access and the ready availability of myriad devices to facilitate this connectivity have catalyzed the transformative role of DCT. By 2023, it is predicted that 5.3 billion individuals worldwide, equating to 66% of the global population, will be internet users, an increase from 3.9 billion in 2018, as indicated by the global IT and networking systems brand, Cisco [2-4].

However, this digital expansion has concurrently led to an alarming escalation in cyber-attacks targeting computer networks, systems, and critical infrastructure. Such attacks can result in significant disruptions to business operations, theft of confidential information and intellectual property, corruption of classified data, and considerable financial loss [5]. The complexity and diversity of these threats are evolving, with cybercriminals continuously adapting to changes in cybersecurity protection measures, leading to a wide range of attacks, often culminating in the total shutdown of crucial technology systems [6]. Malware, an umbrella term for any software designed to steal, conceal, or gain unauthorized access to organizational data or systems, is one of the primary tools employed in these attacks. Other forms of attacks include phishing, denial of service (DoS) and distributed DoS, identity theft, and botnets, as identified in ref. [6]. The escalating prevalence and complexity of these threats necessitate the implementation of robust and reliable intrusion detection systems [7].

An intrusion detection system (IDS) forms a critical component of any security architecture. It functions to proactively monitor, identify, and classify intrusions and attacks on networks and network nodes [7]. Unlike many security solutions such as firewalls, data encryption, and user authentication, IDSs can distinguish between normal and malicious traffic by performing a detailed analysis of network traffic [8]. IDSs can be deployed across individual computers in a network, a network, or multiple network segments to scan network packets and alert network administrators of detected malicious traffic [9].

Machine learning, a subset of artificial intelligence that employs computational and mathematical methods to learn from and make sense of empirical data, has found significant application in cybersecurity, particularly within the realm of intrusion detection. As the volume of data worldwide is expected to grow exponentially, reaching a staggering 175 zettabytes by 2025 (up from 33 zettabytes in 2018), machine learning models hold the potential to learn patterns from this data and make intelligent predictions [10, 11].

Over the past two decades, the integration of machine learning in enhancing the detection of exploits on computer networks has received considerable research attention [12]. With the increasing sophistication of network attacks, machine learning-based IDSs can yield satisfactory detection of anomalous network traffic when trained with high-quality and updated IDS datasets [11].

However, as threats and attacks become increasingly sophisticated, the need for reliable mechanisms to detect malicious traffic has never been more urgent. While several works have utilized deep learning methods to detect network attacks, the reliance of these models on outdated datasets compromises their efficiency in dealing with emerging threats [12].

Hence, the development of detection systems using recent datasets is imperative. Moreover, the integration of multiple datasets would enrich the IDS model with more features, further enhancing its effectiveness. This study, therefore, focuses on the development of a deep learning-based intrusion detection system trained using two contemporary and popular datasets – CICIDS2017 and the Coburg Intrusion Detection Dataset (CIDDS).

The primary contribution of this research lies in the curation of multiple intrusion detection datasets for the development of a deep learning-based intrusion detection system. The implications of these findings are significant, offering practical applications for the development of more effective intrusion detection systems using deep learning techniques.

The remainder of the paper is organized as follows: Section 1 outlines the classification of IDSs and the IDS datasets. Section 2 discusses related works. The methodology implemented and the experimental setups are discussed in Section 3. The research results are presented and discussed in Section 4. The paper concludes with Section 5.

1.1 Classification of intrusion detection systems

The landscape of computer security research has yielded a variety of classification methods for Intrusion Detection Systems (IDS). As depicted in Figure 1, these categorization techniques bifurcate into two primary groups: data-source based techniques and detection-based techniques [13].

Data-source-based techniques are further subdivided into two categories: host-based IDS and network-based IDS. Host-based IDS provide protection to individual computer systems by monitoring all in-system activities and both inbound and outbound traffic [14]. This type of IDS scrutinizes system-specific files, such as operating system logs and registers, to identify potential intrusion events and malicious activity. The direct access to system processes and data files grants host-based IDS visibility into the intended outcome of an attempted attack on the system [14].

Network-based IDS, on the other hand, analyze network packets to detect intrusion attempts within the network. This type of IDS is designed to monitor network traffic affecting multiple endpoints within a network segment, thereby providing a holistic view of network activity [15].

Detection-based techniques are further classified into signature-based detection and anomaly detection methods. Signature-based detection represents attack behaviors as patterns or signatures stored in a database. The method involves the comparison of incoming traffic patterns with the stored signatures in the database. If a match is found, the corresponding traffic is flagged as an intrusion. One of the primary advantages of signature-based detection is its low false-positive rate, attributed to its high detection accuracy for known attacks. Furthermore, it offers detailed reporting of attack types and potential causes. However, the method requires regular updates of the signature database and may not be fully effective against novel, or zero-day, exploits [14].

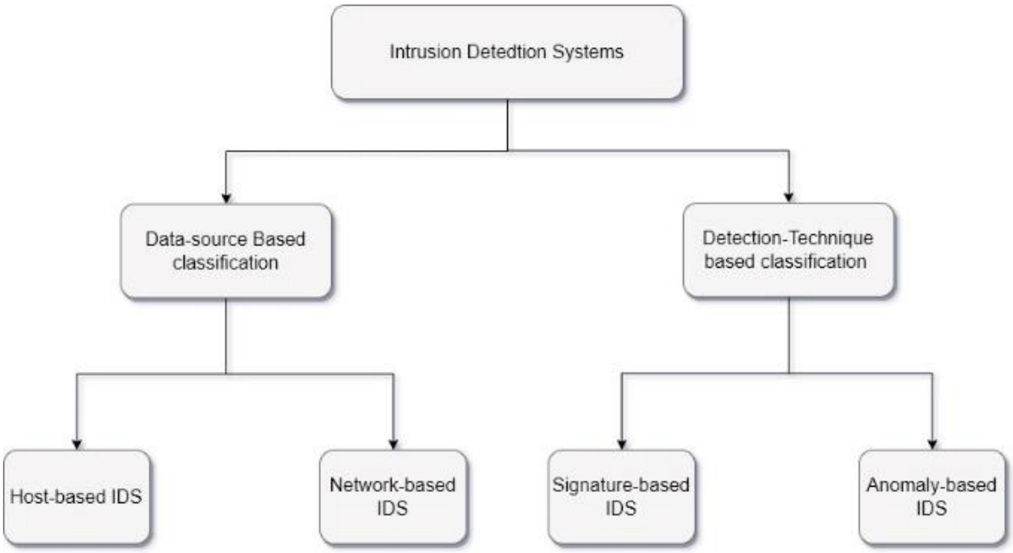

Intrusion Detedtion Systems
Data-source Based classification
Detection-Technique based classification
Host-based IDS
Network-based IDS
Signature-based IDS
Anomaly-based IDS

Figure 1. Intrusion detection systems categorization [13]

In contrast, anomaly-based detection identifies unusual activity by comparing it against a baseline of normal behavior. This method necessitates a detailed understanding of typical network traffic patterns, with any deviation from the norm flagged as a potential intrusion. Anomaly-based IDS entails two main stages: training and testing. During training, the system learns the normal traffic patterns, and in the testing phase, the trained system is exposed to a new dataset for evaluation [16]. The strength of anomaly-based detection lies in its ability to detect zero-day exploits and its resistance to evasion. However, it may result in a higher percentage of false alarms due to its tendency to flag any unrecognized traffic as potentially malicious [16].

1.2 Intrusion detection datasets

In the research and development of Intrusion Detection System (IDS) models, the use of benchmark datasets is pivotal. These datasets serve as the foundation for training models and evaluating their efficacy [12]. Given the data-dependent nature of these models, the quality and comprehensiveness of the datasets significantly impact model performance. For the detection of malicious network traffic, datasets mirroring the behavior of such traffic are imperative for training IDS [11]. This section provides a detailed overview of well-established, as well as recently developed, IDS datasets that are publicly accessible for deep learning model training.

One of the earliest datasets, the DARPA Dataset, was generated in 1998 at the MT Lincoln Laboratory. The dataset, which comprises network-based attacks spanning a seven-week training period and a two-week testing period, is classified into five network traffic categories: Normal, Probe, User to Root attacks, Remote to Local attacks, and DoS attacks [11, 13].

Building upon the DARPA dataset, the KDD99 dataset was developed and has since become the most widely utilized benchmark for IDS. It mirrors the class labels of the DARPA dataset and consists of various property types including basic, content, host-based statistical, and time-based statistical properties. However, redundant and duplicate records present in the KDD99 dataset have been known to cause variation in the results obtained by researchers [11, 17].

Addressing the redundancy issues of the KDD99 dataset, the NSL-KDD dataset was created. By selectively including records from the KDD99 dataset and ensuring balanced classes, the NSL-KDD dataset enables researchers to obtain consistent and comparable results [11, 14].

The UNSW-NB15 dataset, generated by researchers at the University of South Wales, comprises network traffic captured from three virtual servers. The dataset, which was created using four tools-IXIA Perfect-Storm, Tcpdump, Argus, and Bro-IDS-contains a larger array of attacks than the NSL-KDD, including Reconnaissance, Shellcode, Generic, and Worms. It encompasses nearly 2,000,000 vectors and 540,044 features [17].

The Coburg Intrusion Detection Data Set (CIDDS) was developed within an emulated business environment, comprising multiple clients, web servers, and email servers. Intended to serve as a benchmark for anomaly intrusion detection, the CIDDS dataset, like the CICIDS2017 dataset, is flow-based and was generated in an OpenStack virtual environment. It comprises two datasets: CIDDS-001 and CIDDS-002. CIDDS-001 contains unidirectional network traffic collected over a four-week period, featuring DoS attacks, secure shell brute force attacks, and port scan attacks. Conversely, CIDDS-002 is a flow-based dataset solely containing port scan attacks [18].

The CICIDS2017 dataset, proposed by Sharafaldin et al. [19], is often employed for developing anomaly-based IDS. Captured over five days in July 2017, it comprises 80 network features extracted from network traffic using the CICFlowMeter Tool. The dataset includes various attacks: Brute Force SSH and FTP, Denial-of-Service, Heartbleed, Web Attack, Infiltration, Botnet, and Distributed Denial-of-Service [20].

The CSE-CIC-IDS-2018 dataset [21], developed through collaboration between the Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity (CIC), includes seven types of attacks: Heartbleed, Denial-of-Service, Distributed Denial-of-Service, Web Attacks, Brute-force, Botnet, and Infiltration. The dataset was generated from an attack scenario setup comprising fifty machines within an organization that includes four-hundred-and-twenty PCs and thirty servers across five departments. Network parameters were obtained from the captured traffic using the CICFlowMeter application [17].

The CIC-DDoS2019 dataset [22] includes various forms of Distributed DoS attacks executed using OSI application layer protocols such as TCP and UDP. The dataset was generated on two distinct days in 2019, January 12 (training data) and March 11 (test data), and includes twelve types of Distributed Denial-of-Service attacks for training and seven types for testing: SYN, MSSQL, LDAP, UDP, UDP-Lag, PortScan, and NetBIOS [14].

Finally, the LITNET-2020 dataset contains 12 network attacks sourced from servers in four locations within the Lithuanian-wide network. The attacks include Smurf, UDP-Flood, ICMP-Flood, HTTP Flood, TCP SYN-flood, LAND Attack, Blaster Worm, Code Red Worm, Reaper Worm, Spam Bot Detection, Scanning/Spread, and Packet Fragmentation. The dataset was gathered over a ten-month period [14, 23].

2. RELATED WORKS

Tang et al. [24] employed a Deep Neural Network (DNN) trained on the NSL-KDD dataset to constitute an IDS for SDNs. Their DNN model was designed with a six-dimensional input layer, three hidden layers consisting of 12, 6, and 3 neurons respectively, and a two-dimensional output layer. Through two-class classification, the system attained an accuracy of 75.75% with a batch size of 10 and an epoch value of 100.

In a different context, Kang and Kang [25] formulated an IDS for vehicular networks, specifically controller area networks (CAN), using a DNN. The model, trained using unsupervised deep belief network (DBN) pre-training, differentiated normal and malicious packets by learning the statistical properties of network packet data. The system achieved an impressive detection performance average of 98% with a false positive rate of less than one to two percent, demonstrating its efficacy in providing real-time responses to attacks.

Feng et al. [26] developed a plug-and-play device that uses deep learning to detect privacy and DoS attacks in ad hoc networks. The system captures traffic data and sends notifications when attacks are detected. Three deep learning models, DNN, CNN, and LSTM, were used to detect XSS and SQL attacks, with the DNN detecting DoS attacks. The system achieved an accuracy of 98.5%, and the DNN and CNN had detection accuracies of 57% and 78% respectively for XSS attacks, using the KDD CUP 99 dataset.

Kim et al. [27] proposed a DNN trained on the KDD 1999 dataset for detecting constantly evolving network exploits. The model incorporated four hidden layers and 100 hidden units. The Rectified Linear Unit (ReLU) was utilized as the activation function during the model's training, with the stochastic optimization technique. The model demonstrated an accuracy of ninety-nine percent.

For binary and multiclass classification, Yin et al. [28] proposed an IDS using a Recurrent Neural Network (RNN). The system's performance was evaluated at different learning rates and varying numbers of neurons. With a learning rate value of 0.1 and 80 hidden nodes, the binary classification (anomaly detection) performance was reported to be more accurate. Meanwhile, a learning rate of 0.5 and 80 hidden nodes resulted in more accurate multi-class attack detection performance. Comparisons with other machine learning models, such as naive Bayes, RF, and SVM, indicated the superiority of the proposed system.

In the realm of SDNs, Tang et al. [29] proposed an IDS that implements a combination of Gated Recurrent Unit (GRU) and RNN. Using the NSL-KDD dataset for training, the GRU-RNN IDS achieved an accuracy performance of 89% without deteriorating network performance.

Similarly, Jiang et al. [30] proposed a multi-channel IDS that uses Long Short-Term Memory (LSTM) and RNN. The performance of this attack detection system was evaluated using the NSL-KDD dataset, with the LSTM-RNN IDS achieving a detection rate of 99.23%, a false positive value of 9.86%, and an accuracy value of 98.94%.

In a different approach, Sharafaldin [31] used flow-based features from the CIC-DDoS2019 dataset. An RNN was used to alleviate the data information loss caused by sequential traffic, achieving an average AUC of 0.988 in the majority of the tests. However, the study also noted that the strategy might not be suitable in situations where the network data has a high dependence on time due to the limitations of RNNs.

Nasr et al. [32] designed DeepCorr, a system that uses a Convolutional Neural Network (CNN) to perform flow correlation on Tor traffic, outperforming existing solutions with higher accuracy. The CNN architecture of DeepCorr consists of two convolution layers and three fully connected neural network layers. The model was trained with the UMASS dataset generated in 2018 and achieved a false positive rate of 0.1% and a true positive rate of 80% with a learning rate value of 0.0001.

To further enhance network security, Hu et al. [33] and Okokpujie et al. [34] developed an IDS that combines the ADASYN algorithm and an improved CNN. The ADASYN algorithm was used to balance the distribution of minority data samples to prevent model bias. The improved CNN increased feature diversity and reduced the effect of interchannel information redundancy on model training. When the improved CNN model was trained with the NSIn the current literature, various research efforts have been dedicated to the development of Intrusion Detection Systems (IDS) utilizing deep learning techniques. These systems play a critical role in fortifying the security of diverse network types, ranging from Software-Defined Networks (SDNs) to vehicular and ad hoc networks.

3. METHODOLOGY

The conceptual research framework for developing a malicious network traffic detection system is shown in Figure 2. The conceptual framework follows a standard research pipeline based on the knowledge acquired from the examination of research literature on deep learning model utilization.

Problem formulation. The first stage in the proposed framework is problem formulation. This describes the aim of the proposed system, which is to develop a network malicious detection system. The objective intended to be examined is the possibility of developing a malicious traffic detection system by utilizing two datasets for the model training process and examining the performance of the trained model. As discovered from reviewed works of literature, previously developed network intrusion detection systems are based on old attacks in popular datasets [14]. This greatly reduces the performance of such systems against an ever-growing threat landscape. In addition, because the publicly available datasets are limited in the number of attacks traffic captured, this research seeks to combine two recent datasets with various attacks captured, to have a system that is resilient to a wider range of attacks.

Data collection. To develop a deep learning model, relevant and quality data is required. The two datasets that were used in this research work are the CICIDS2017 dataset and the CIDDS dataset. They are both flow-based and generated in emulated environments. CICIDS2017 contains the following attacks: Botnet, DoS, DDos, Heartbleed, Infiltraion, Brute force, Port Scan, and Web Attack; while CIDDS, which have two sets of datasets generated by the same authors (CIDDS) contains the following attacks: DoS, Port Scan, Brute Force, Scan attacks.

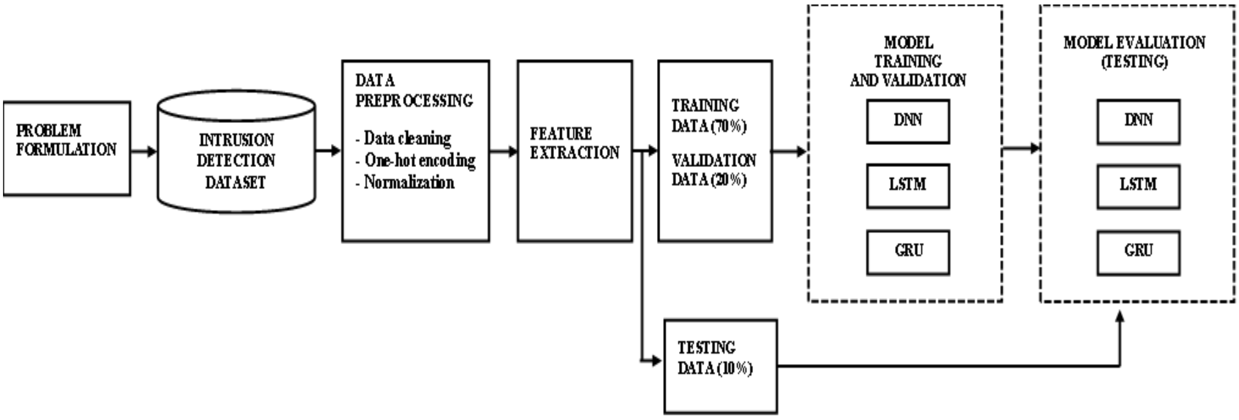


Figure 2. Proposed research conceptual framework

Table 1. Summary of the combined dataset

S/N	Traffic Label	Heading 3
1	Normal traffic	14,485,522
2	DoS	2,002,167
3	Scan	568,730
4	Port Scan	424,842
5	DdoS	128,027
6	Brute Force	18,827
7	Bot	1,966

Table 2. Features of the combined dataset

S/N	Feature Name
1	Duration
2	Source Port
3	Destination Port
4	Protocol (TCP, UDP, ICMP, IGMP)
5	Packets
6	Bytes
7	Urgent Flag
8	Acknowledge Flag
9	Push Flag
10	Reset Flag
11	Finish Flag
12	Attack Label

Table 3. Model parameters

S/N	Parameters	Value
1	Number of Features	14
2	Drop out	0.1
3	Optimizer	‘adam’
4	Loss Function	Categorical cross-entropy
5	Number of Hidden Layers	3
6	Layer Configuration	(256, 128,128), (128, 64, 64)
7	Metrics	Accuracy, AUC, Precision, True Positives False Positives, True Negatives, False Negatives
8	Batch Size	256, 512
9	Epochs	30
10	Output Layer Activation Function	Softmax
11	Hidden Layers Activation Function	ReLU

Data pre-processing and feature extraction. The following specific set of activities were carried out during this stage: data cleaning, one-hot encoding of categorical values, and data normalization. In the data cleaning stage, we dropped rows with empty values; represented the feature in the appropriate formats; selected the relevant columns; dropped attacks (Heartbleed, infiltration, web attack- bruteforce, cross-site scripting and SQL Injection) with very little number of samples to eliminate the effect of data imbalance; and merged the two datasets along the appropriate axis (summary of merged data is shown in Table 1). The summary of the data features is contained in the Table 2. In next stage, one-hot encoding was carried out on the protocol and attack label column. The protocol column has four unique categorical values (TCP, UDP, ICMP, and IGMP), while the attack label column has seven unique categorical values. However, to prevent high correlation among the protocols, the generated column for ICMP was dropped. Finally, in last stage, data normalization was done to place every numerical value with the internal of 0 and 1, using min-max normalization. After pre-processing the dataset, it was then split into training, validation, and testing sets with ration of 70:20:10 respectively.

Model selection, configuration, training, and validation. Three deep learning models were used in this research because of performance and popularity in previous works, which are Deep Neural Network (DNN), Long Short-Term Memory (LSTM), and Gate Recurrent Unit (GRU). The parameters used for the configuration of the models are detailed in Table 3. Two experimental setups are utilized for training, validating, and testing the models. The configurations implemented are detailed in Table 4. After configuring the models, the training set as well as validation set were used to carryout training and validation of the three deep learning models. The models were trained with two batch sizes (512 and 256) with a Dropout value of 0.1 after the first and second hidden layer for 30 epochs.

Table 4. Experimental setups

S/N	Experimental Setup One		Experimental Setup Two	
1	Parameter	Value	Parameter	Value
2	Input Layer	14	Input layer	14
3	Hidden Layer 1 (Neurons & activation function)	**256 (ReLU)**	Hidden Layer 1 (Neurons & activation function)	**128 (ReLU)**
4	Dropout	0.1	Dropout	0.1
5	Hidden Layer 2 (Neurons & activation function)	**128 (ReLU)**	Hidden Layer 2 (Neurons & activation function)	**64 (ReLU)**
6	Dropout	0.1	Dropout	0.1
7	Hidden Layer 3 (Neurons & activation function)	**128 (ReLU)**	Hidden Layer 3 (Neurons & activation function)	**64 (ReLU)**
8	Output	7 (Softmax)	Output	7 (Softmax)
9	Batch size	**512**	Batch size	**256**
10	Epochs	30	Epochs	30

4. RESULTS AND DISCUSSION

After carrying out model training and validation, we evaluate the performance of the model using standard performance metrics such as accuracy, precision, recall, and F1-score (using Eqs. (1)-(4)) [35, 36].

$$Precision = \frac{TP}{TP + TN} \tag{1}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

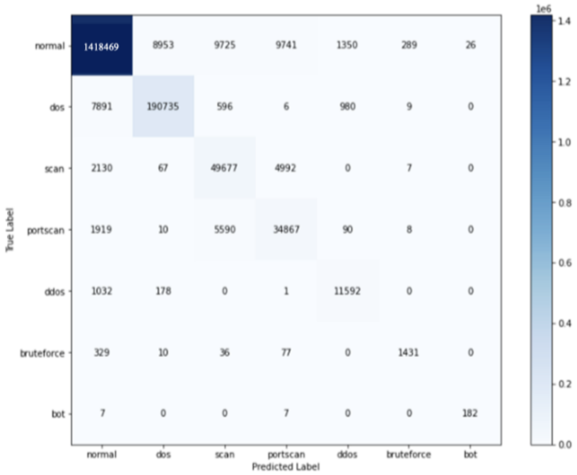
$$F1 - Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{4}$$

where, *TP* – True Positive; *TN* – True Negative; *FP* – False Positive; *FN* – False Negative.

The result of the training process of the three models is contained in the Table 5, and it shows that, across the three models for the two experimental setups, LSTM achieve the best performance, particularly in experimental setup one, with an accuracy of 98.09%, precision of 98.14%, F1-score of 98.09%, and recall of 98.05%.

Table 5. Model performance result

	Metrics	DNN	LSTM	GRU
Experimental Setup One	Accuracy	97.68%	**98.09%**	97.91%
	Precision	97.81%	**98.14%**	97.96%
	F1-Score	97.70%	**98.09%**	97.91%
	Recall	97.60%	**98.05%**	97.87%
Experimental Setup Two	Accuracy	97.54%	**97.97%**	97.80%
	Precision	97.69%	**98.01%**	97.86%
	F1-Score	97.55%	**97.97%**	97.80%
	Recall	97.42%	**97.93%**	97.74%



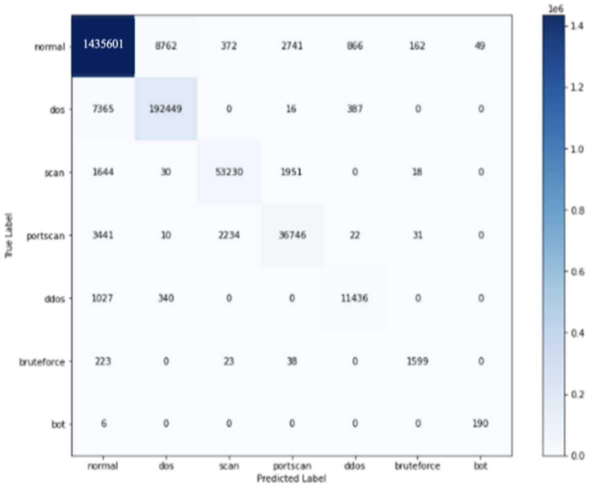
	normal	dos	scan	portscan	ddos	bruteforce	bot
normal	1418469	8953	9725	9741	1350	289	26
dos	7891	190735	596	6	980	9	0
scan	2130	67	49677	4992	0	7	0
portscan	1919	10	5590	34867	90	8	0
ddos	1032	178	0	1	11592	0	0
bruteforce	329	10	36	77	0	1431	0
bot	7	0	0	7	0	0	182

Figure 3. Confusion matrix of DNN (DNN)

After the models were trained and validated, the test dataset (10%) was used to test the developed models. The classification reports obtained (for the experimental setups) are presented in Table 6 and Table 7. The outcome of the test shows that, across the three models, the weighted average value for the precision, recall, and F1-score are generally higher (DNN1 and DNN2 - 0.97; LSTM 1 and LSTM 2 - 0.98; GRU 1 and GRU 2 - 0.98 and 0.97 respectively) than the macro average for the three models. The results obtained for the macro average of the three models in the two experimental setups are satisfactory. This is because, while the weighted average is computed by factoring the percentage of every class label in the dataset, which does not favor the minority class; the macro average is computed without taking into consideration the percentage of each class, thus giving a true picture of the model performance across all the classes.

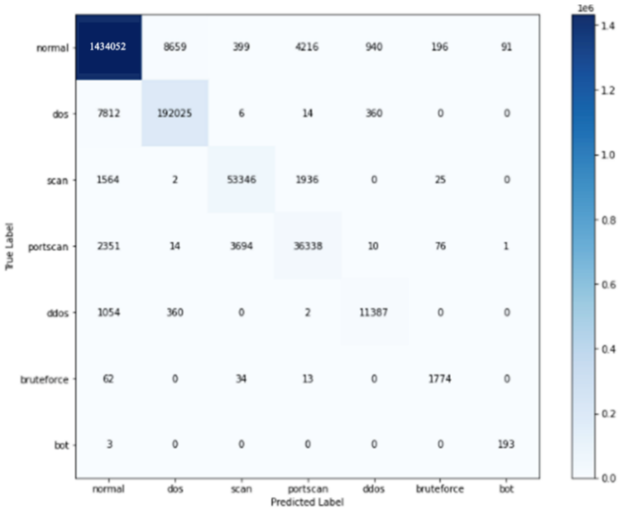
The confusion matrixes for the testing of the models are depicted in Figures 3-5, which show the classification result. However, it can be observed from the three confusion matrixes that a significant number of portscan attack was classified as scan attack and vice versa. Other classes however achieved a higher detection rate.

Comparing the outcome of this research with some other works from literature indicates that the LSTM model configured and trained using experimental setup one achieved a better performance as presented in the Table 8.



	normal	dos	scan	portscan	ddos	bruteforce	bot
normal	1435601	8762	372	2741	866	162	49
dos	7365	192449	0	16	387	0	0
scan	1644	30	53230	1951	0	18	0
portscan	3441	10	2234	36746	22	31	0
ddos	1027	340	0	0	11436	0	0
bruteforce	223	0	23	38	0	1599	0
bot	6	0	0	0	0	0	190

Figure 4. Confusion matrix of LSTM



	normal	dos	scan	portscan	ddos	bruteforce	bot
normal	1434052	8659	399	4216	940	196	91
dos	7812	192025	6	14	360	0	0
scan	1564	2	53346	1936	0	25	0
portscan	2351	14	3694	36338	10	76	1
ddos	1054	360	0	2	11387	0	0
bruteforce	62	0	34	13	0	1774	0
bot	3	0	0	0	0	0	193

Figure 5. Confusion matrix of GRU

Table 6. Classification report of the testing of the models (setup one)

		DNN			LSTM			GRU		
	Precision	Recall	f1-Score	Precision	Recall	f1-Score	Precision	Recall	f1-Score	Support
Normal	0.99	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99	1448553
DoS	0.95	0.95	0.95	0.95	0.96	0.96	0.96	0.96	0.96	200217
Scan	0.76	0.87	0.81	0.95	0.94	0.94	0.93	0.94	0.93	56873
Portscan	0.70	0.82	0.76	0.89	0.86	0.88	0.85	0.86	0.85	42484
DDoS	0.83	0.91	0.86	0.90	0.89	0.90	0.90	0.89	0.89	12803
Bruteforce	0.82	0.76	0.79	0.88	0.85	0.87	0.86	0.94	0.90	1883
Bot	0.88	0.93	0.90	0.79	0.97	0.87	0.68	0.98	0.80	196
Macro Average	0.85	0.89	0.87	0.91	0.92	0.91	0.88	0.94	0.90	
Weighted Average	0.97	0.97	0.97	0.98	0.98	0.98	0.98	0.98	0.98	
Total Support										1763009

Table 7. Classification report of the testing of the models (setup two)

		DNN			LSTM			GRU		
	Precision	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**	**Precision**	**Recall**	**f1-Score**	**Support**
Normal	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1448553
DoS	0.95	0.95	0.95	0.95	0.96	0.96	0.95	0.95	0.95	200217
Scan	0.87	0.87	0.87	0.92	0.91	0.92	0.86	0.79	0.82	56873
Portscan	0.77	0.78	0.77	0.81	0.86	0.83	0.69	0.74	0.71	42484
DDoS	0.81	0.93	0.87	0.86	0.91	0.88	0.86	0.85	0.85	12803
Bruteforce	0.90	0.70	0.79	0.61	0.69	0.65	0.80	0.80	0.80	1883
Bot	0.73	0.99	0.84	0.84	0.92	0.88	0.75	0.39	0.52	196
Macro Average	0.86	0.89	0.87	0.86	0.89	0.87	0.84	0.79	0.81	
Weighted Average	0.98	0.97	0.97	0.98	0.98	0.98	0.97	0.97	0.97	
Total Support										1763009

Table 8. Comparison of obtained result with results of earlier literature

S/N	Ref.	Deep Learning Model	Number of Dataset Used	Dataset	Results
1	This work	LSTM	3	CIDDS-001, CIDDS-002, CICIDS 2017	Precision: 98.14% Accuracy: 98.09% F1-Score: 98.09% TPR: 98.05%; TNR: 99.69%; FPR: 0.31%; FNR: 1.95%
2	[37]	LSTM	1	CIDDS-001	Accuracy: 99.91 % Precision: 98.37 % TPR: 71.40 % F1-Score: 74.23 %
3	[38]	DNN	1	CICIDS 2017	Accuracy: 97.73 %
4	[39]	LSTM	1	CIDDS-001	Accuracy: 84.83 % Precision: 85.14 % Recall: 88.34 % FPR: 17.22 %

5. CONCLUSION

In this research work, a deep learning-based intrusion detection system was developed. The chosen deep learning models were trained using two recent benchmark datasets (CICIDS 2017 and Coburg Intrusion Detection Data Sets (CIDDS)) for intrusion detection. The two datasets were preprocessed and merged on common and essential features. The combined dataset was then used to train the deep-learning models. The deep learning models were configured using two sets of experimental setups. The LSTM trained using three hidden layers with 256, 128, and 128 neurons in Layers 1, 2 and 3, respectively, and with a batch size of 512, achieved the best performance in all the metrics, with an accuracy of 98.09% and precision of 98.14 %. The implications of creating a deep-learning-based intrusion detection system with two or more datasets include a more comprehensive and robust training process. The system can learn to detect different types of intrusions and attacks using multiple datasets, resulting in a more accurate and reliable detection system. Combining datasets with common features can also improve the system's ability to detect patterns and behaviors associated with attacks, resulting in a better understanding of security threats while also resulting in a more generalized model that can be applied to a variety of scenarios. Ultimately, using multiple datasets can improve system performance, effectiveness, and adaptability to various security applications.

The major challenge experienced, however, was data imbalance. The CICIDS 2017 dataset had some attack samples with less than fifty samples (i.e., eleven, twenty-one, and thirty-six samples for Heartbleed, web attack – SQL Injection, and Infiltration), while some were less than one thousand samples (i.e., six hundred and fifty-two samples for web attack – Cross Site Scripting attack). These insufficient attack samples were removed in order to eliminate the impact of the data imbalance since these data are not enough for deep learning experimentation. Therefore, as possible in future work, data augmentation techniques can be applied to address the imbalance of some attack samples in the dataset. This can take the form of creating more attacks from the ones available or by creating more samples by simulating more of such attacks.

ACKNOWLEDGMENT

The authors acknowledge the part sponsorship of the Covenant University Centre for Research, Innovation, and Discovery (CUCRID), Ota, Ogun State, Nigeria.

REFERENCES

[1] AlDaajeh, S., Saleous, H., Alrabaee, S., Barka, E., Breitinger, F., Raymond Choo, K.K. (2022). The role of national cybersecurity strategies on the improvement of cybersecurity education. Computers & Security, 119: 102754. https://doi.org/10.1016/J.COSE.2022.102754

[2] Garg, S., Singh, A., Kaur, K., Aujla, G.S., Batra, S., Kumar, N., Obaidat, M.S. (2019). Edge computing-based

security framework for big data analytics in VANETs. IEEE Network, 33(2): 72-81. https://doi.org/10.1109/MNET.2019.1800239

[3] Singh, A., Garg, S., Batra, S., Kumar, N., Rodrigues, J.J.P.C. (2018). Bloom filter based optimization scheme for massive data handling in IoT environment. Future Generation Computer Systems, 82: 440-449. https://doi.org/10.1016/J.FUTURE.2017.12.016

[4] Aujla, G.S., Chaudhary, R., Kaur, K., Garg, S., Kumar, N., Ranjan, R. (2019). SAFE: SDN-assisted framework for edge-cloud interplay in secure healthcare ecosystem. IEEE Transactions on Industrial Informatics, 15(1): 469-480. https://doi.org/10.1109/TII.2018.2866917

[5] Byrne, M.D. (2021). Cybersecurity and the new age of ransomware attacks. Journal of PeriAnesthesia Nursing, 36(5): 594-596. https://doi.org/10.1016/j.jopan.2021.07.004

[6] European Union Agency for Cybersecurity (ENISA). Insider threat - ENISA Threat Landscape. https://www.enisa.europa.eu/.

[7] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7: 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334

[8] Kilincer, I.F., Ertam, F., Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188: 107840. https://doi.org/10.1016/j.comnet.2021.107840

[9] Gamage, S., Samarabandu, J. (2020). Deep learning methods in network intrusion detection: A survey and an objective comparison. Journal of Network and Computer Applications, 169: 102767. https://doi.org/10.1016/J.JNCA.2020.102767

[10] Mukkamala, S., Janoski, G., Sung, A. (2002). Intrusion detection using neural networks and support vector machines. In Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), Honolulu, HI, USA, pp. 1702-1707. https://doi.org/10.1109/IJCNN.2002.1007774

[11] Liu, H.Y., Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9(20): 4396. https://doi.org/10.3390/app9204396

[12] Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R.C., Bellekens, X. (2018). A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. https://www.researchgate.net/publication/325709382_A_Taxonomy_and_Survey_of_Intrusion_Detection_System_Design_Techniques_Network_Threats_and_Datasets.

[13] Lee, S.W., Sidqi, H.M., Mohammadi, M., Rashidi, S., Rahmani, A.M., Masdari, M., Hosseinzadeh, M. (2021). Towards secure intrusion detection systems using deep learning techniques: Comprehensive analysis and review. Journal of Network and Computer Applications, 187: 103111. https://doi.org/10.1016/J.JNCA.2021.103111

[14] Macas, M., Wu, C., Fuertes, W. (2022). A survey on deep learning for cybersecurity: Progress, challenges, and opportunities. Computer Networks, 212: 109032. https://doi.org/10.1016/j.comnet.2022.109032

[15] Grance, T., Stevens, M., Myers, M. (2003). Guide to Selecting Information Technology Security Products. NIST Special Publication, Gaithersburg, MD. https://doi.org/10.6028/NIST.SP.800-36

[16] Kilincer, I.F., Ertam, F., Sengur, A. (2021). Machine learning methods for cyber security intrusion detection: Datasets and comparative study. Computer Networks, 188: 107840. https://doi.org/10.1016/J.COMNET.2021.107840

[17] Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50: 102419. https://doi.org/10.1016/j.jisa.2019.102419

[18] Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A. (2017). Flow-based benchmark data sets for intrusion detection. In European Conference on Information Warfare and Security, ECCWS, pp. 361-369.

[19] Sharafaldin, I., Lashkai, A.H., Ghorbani, A.A. (2018). Intrusion detection evaluation dataset (CIC-IDS2017). Canadian Institute for Cybersecurity. https://www.unb.ca/cic/datasets/ids-2017.html, accessed on Jul. 19, 2022.

[20] Moustafa, N., Turnbull, B., Choo, K.K.R. (2019). An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things. IEEE Internet of Things Journal, 6(3): 4815-4830. https://doi.org/10.1109/JIOT.2018.2871719

[21] Sperotto, A., Sadre, R., Van Vliet, F., Pras, A. (2009). A Labeled Data Set for Flow-Based Intrusion Detection. In: Nunzi, G., Scoglio, C., Li, X. (eds) IP Operations and Management. IPOM 2009. Lecture Notes in Computer Science, vol 5843. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04968-2_4

[22] Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, pp. 1-8. https://doi.org/10.1109/CCST.2019.8888419

[23] Damasevicius, R., Venckauskas, A., Grigaliunas, S., Toldinas, J., Morkevicius, N., Aleliunas, T., Smuikys, P. (2020). LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. Electronics, 9(5): 800. https://doi.org/10.3390/electronics9050800

[24] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M. (2016). Deep learning approach for network intrusion detection in software defined networking. In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), Fez, Morocco, pp. 258-263. https://doi.org/10.1109/WINCOM.2016.7777224

[25] Kang, M.J., Kang, J.W. (2016). Intrusion detection system using deep neural network for in-vehicle network security. PLoS One, 11(6): e0155781. https://doi.org/10.1371/journal.pone.0155781

[26] Feng, F., Liu, X., Yong, B., Zhou, R., Zhou, Q. (2019). Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device. Ad Hoc Networks, 84: 82-89. https://doi.org/10.1016/j.adhoc.2018.09.014

[27] Kim, J., Shin, N., Jo, S.Y., Kim, S.H. (2017). Method of intrusion detection using deep neural network. In 2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017, pp. 313-316.

https://doi.org/10.1109/BIGCOMP.2017.7881684

[28] Yin, C., Zhu, Y., Fei, J., He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5: 21954-21961. https://doi.org/10.1109/ACCESS.2017.2762418

[29] Tang, T.A., Mhamdi, L., McLernon, D., Zaidi, S.A.R., Ghogho, M. (2018). Deep recurrent neural network for intrusion detection in SDN-based networks. In 2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), Montreal, QC, Canada, pp. 202-206. https://doi.org/10.1109/NETSOFT.2018.8460090

[30] Jiang, F., Fu, Y.S., Gupta, B.B., Liang, Y.S., Rho, S., Lou, F., Meng, F.Z., Tian, Z.H. (2020). Deep learning based multi-channel intelligent attack detection for data security. IEEE Transactions on Sustainable Computing, 5(2): 204-212. https://doi.org/10.1109/TSUSC.2018.2793284

[31] Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In Proceedings - International Carnahan Conference on Security Technology, Chennai, India. https://doi.org/10.1109/CCST.2019.8888419

[32] Nasr, M., Bahramali, A., Houmansadr, A. (2018). DeepCorr: Strong flow correlation attacks on tor using deep learning. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1962-1976. https://doi.org/10.1145/3243734.3243824

[33] Hu, Z.Q., Wang, L.J., Qi, L., Li, Y.M., Yang, W.Z. (2020). A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network. IEEE Access, 8: 195741-195751. https://doi.org/10.1109/ACCESS.2020.3034015

[34] Okokpujie, K., Mughole, D., Badejo, J.A., Adetiba, E. (2022). Congestion intrusion detection-based method for controller area network bus: A case for KIA SOUL vehicle. Mathematical Modelling of Engineering Problems, 9(5): 1298-1304. https://doi.org/10.18280/mmep.090518

[35] Okokpujie, K., Kennedy, G.C., Nzanzu, V.P., Molo, M.J., Adetiba, E., Badejo, J. (2021). Anomaly-based intrusion detection for a vehicle can bus: A case for Hyundai Avante CN7. Journal of Southwest Jiaotong University, 56(5). https://doi.org/10.35741/issn.0258-2724.56.5.14

[36] Okokpujie, K., Kennedy, C.G., Nnodu, K., Noma-Osaghae, E. (2023). Cybersecurity awareness: Investigating students' susceptibility to phishing attacks for sustainable safe email usage in academic environment (a case study of a Nigerian leading university). International Journal of Sustainable Development and Planning, 18(1): 255-263. https://doi.org/10.18280/ijsdp.180127

[37] Oliveira, N., Praça, I., Maia, E., Sousa, O. (2021). Intelligent cyber attack detection and classification for network-based intrusion detection systems. Applied Sciences, 11(4): 1674. https://doi.org/10.3390/app11041674

[38] Faker, O., Dogdu, E. (2019). Intrusion detection using big data and deep learning techniques. Proceedings of the 2019 ACM Southeast Conference, pp. 86-93. https://doi.org/10.1145/3299815.3314439

[39] Althubiti, S.A., Jones, E.M., Roy, K. (2018). LSTM for anomaly-based network intrusion detection. 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, pp. 1-3. https://doi.org/10.1109/ATNAC.2018.8615300