Journal of Engineering Science and Technology
Vol. 16, No. 1 (2021) 711 - 727

DEEP-INTRUSION DETECTION SYSTEM WITH ENHANCED UNSW-NB15 DATASET BASED ON DEEP LEARNING TECHNIQUES

A. M. ALEESA[1,*], MOHAMMED YOUNIS[2], AHMED A. MOHAMMED[3], NAN M. SAHAR[1]

[1]Department of Electrical & Electronic Engineering, UTHM, Parit Raja, Johor Baru, 86400, Malaysia
[2]Department of Electrical Engineering, University of Mosul, Mosul, Ninevah, 41002, Iraq
[3]Department of computer and information Engineering, Ninevah university, Mosul, Ninevah, 41002, Iraq
*Corresponding Author: Ahm3d.aleesa@gmail.com

Abstract

Growth in the number of devices and data has raised serious security concerns, that have increased the importance of the development of advanced intrusion detection systems (IDS). Deep learning can handle big data and in various fields has shown a great performance. Consequently, security specialists are aiming to adopt deep learning in an intrusion detection system. Numerous studies have been done on this topic which have led to many different approaches. Most of these approaches use predefined features extracted by an expert in order to classify network traffic. In addition, UNSW-NB15 dataset was developed in different separated files and labelled based on binary classification, in this research, we aim to merge the whole dataset to be in one file so it can test models once, instead of test models separately for each file. then used attacks families in the dataset as new label so that it will develop multi-classification labelled dataset. We investigated the performance of deep learning with the enhanced dataset, within two classification categories (Binary and Multi-Class). We compared our proposed deep learning model results with related works. We have used accuracy and loss to evaluate the efficiency of deep learning and machine learning models in the enhanced dataset. Our proposed Deep learning models Performed yielded accuracy of 99.59% in multi-class classification and 99.26% in binary classification.

Keywords: ANN, Deep learning, DNN, Intrusion detection system, Machine learning, UNSW-NB15.

1. Introduction

Over the past decades, all aspects of our lives have been exposed out to the Internet. Experts predict that 50 billion connected devices will be usable by 2020 [1, 2]. The difficulty of safeguarding networks and preventing security threats grow as infrastructure becomes more interconnected. Over the years, the vulnerabilities of banking systems, healthcare systems, and IoT tools have increased. These attacks annually lead to billions of dollars in losses in addition to system damage at critical periods. In cybersecurity, particularly in intrusion detection systems, has led to higher importance [3]. One of the related problems with most new infrastructures is that security data specifications are often a backdrop. The result of any machine learning algorithm applied is expected to be affected, but an experiment to assess the discrepancies is still to be seen. The result of any machine learning algorithm applied toward a problem is believed to be affected, however, an analysis to analyse the variations needs to be seen [4, 5].

A network intrusion detection (NIDS) application monitoring network traffic for malevolent purposes is a software application [6]. The unique common approach is to check anomalies in a net's activities or all other than usual net behaviour. Detection of abnormalities creates model behaviors of networks and other devices and then looks for abnormal behaviour patterns at a much faster pace. Machine learning is used to develop a model for anomaly detection and two approaches are deep learning and shallow learning. The technologies used for the prediction model are mostly determined by the Shallow learning models which mostly depend on the features. Conversely, Deep learning (DL) models can extract improved representatives from the raw data to generate many improved models. Because they are made of multiple layers, deep learning can understand even faster. in comparison with shallow learning which does not consist of hidden layers, the DL model can extract a better representation of the feature at each layer. In general, DDoS attacks are divided into attacks of an infrastructure layer and attacks of an application layer. Attacks in the network layer include protocol attacks and Volume-based attacks [7]. Protocol attacks are focused on draining victims related communication devices or server resources. Application layer attacks aim to disable a web server. Volume-based attacks are focused on sending enormous amounts of traffic to the bandwidth of the victims. very little knowledge or skill is required to launch volume attacks where it is the most common attack among all DDoS attacks.

Based on Avital et al. [8] “2019 Global DDoS Threat Landscape Report”, The study analyses 3,643 DDoS attacks in the network layer over the entire 2019 and 42,290 DDoS attacks in the application layer which Imperva has mitigated between May and December 2019. In April, a network layer DDoS-attack hit 580 million packets per second (PPS) and a separate application layer-attack lasting 13 days and peaking at 292,000 requests per second (RPS) was reported as one of the biggest Network and application layer attacks ever. These large-scale attacks, though, remained above practice. Overall, we have seen smaller, quicker, and more frequent threats. This pattern may mean that attackers are trying to wreak havoc before the mitigation program begins, but for Imperva, it is not appropriate where time is almost nil to mitigate. The most important volumetric network layer DDoS attack (UDP / ICMP flood) we saw in Imperva's corporate website was found and instantly mitigated in January this year. It was a 92Gbps threat with 10.38 million packets per second [8].

in this contest, our contribution is to improve UNSW-NB15 dataset to be used with deep learning because old machine learning techniques would take a very long time and the size of the dataset would not affect the performance of machine learning techniques, whereas the size of the used dataset would affect the performance of deep learning techniques [9]. The current dataset is built as four separated CSV files, in order to use the dataset as full set and contain all types of attacks at once, we will merge this dataset to become a one csv file. Moreover, the dataset will need to be pre-processed in order to develop it as one file. Additionally, the dataset used a binary labelled classification, so that the second contribution will improve labelling of the dataset to be multi-class based on the attacks' families beside the normal packets which will give us 10 classes. Accordingly, we will deploy deep learning models on the improved dataset, to find the Impact of labelling types on the classification accuracy for deep learning techniques as an intrusion detection system.

2. Related Work

One classifier is known as an Incremental Learning Algorithm based on a support vector machine (SVM), which has been proposed by Myint and Meesad [10]. A prediction is made with SVM and reduces the steps necessary for the complexity and calculation of the algorithm, time to train the data set repeatedly, and error set. To check the performance of the system using a data set named KDD Cup99. 41 features of the incoming data set may well be predicted by the proposed system. Farnaaz and Jabbar [11] have proposed a model using the intrusion detection RF classifier was introduced. Therefore, RF is used as an ensemble classifier and the model does more than other standard classifiers to distinguish attacks. For the implementation, NSL-KDD is used as a dataset, and with a high detection rate and low false alarm rate, the proposed model is successful [12].

Peddabachigari et al. [13] has proposed STL-IDS an efficient deep learning approach to promoting the process of self-learning. Chowdhuri et al. suggested the framework [14] can be used both for interactive learning and the reduction of size. In this method, preparation and testing time was needed to achieve greater prediction precision for SVM. The current solution increases the detection of network intrusion. Moreover, Chandre et al. [15], The intrusion detection decision tree has been tested. DARPA dataset from 1998 tested intrusion detection with the decision and the system has better accuracy than conventional models. Once more, the findings show that test times and training times are higher than the SVM. Muna et al. [16] have proposed NIDS based on Naïve Bayes framework.

The assigned method has been identified to deliver better efficiency in terms of false-positive rates, computational times, and prices for implementation of KDD Cup 99 as a dataset. Muna et al. [16] proposed an ICS-dependent irregularity navigation system for IICSs that can learn and accept data from the TCP / IP packs. This requires a sequential preparation phase using a feed-forward neural system and deep automation encoder and a detailed input system development to be tested by two data sets bases, NSL-KDD and UNSW-NB15 that are sure to be understood. Since the test results show that this approach can achieve a higher detection rate and a less false positive rate than the other eight methods, genuine IICS conditions can be revised [16].

Shi et al. [17] proposed another approach to simplification based on in-depth training and features selection processes to offer the ideal and highlights for traffic arrangements. Furthermore, in [18] symmetrical vulnerability is exploited to remove the irrelevant highlights in the data set of system traffic. A feature age reveals, based on deep learning and is related to these relevant dimensional lowering features and selection period. Finally, Weighted Symmetric Complexity is exploited by expelling repeated ones to pick the optimal features. Because of real traffic flows, exploratory findings show that the approach suggested can only decrease the aspect of the highlight, but also resolve the negative effects on both the ml techniques of multi-class lop-sidedness and idea float [19].

In the context of IDS research and development, the following ML techniques have been extensively used: Random Forest (RF) [20], k-Nearest Neighbours (kNN) [21], (SVM) [22, 23], Decision Tree (DT) [24], Artificial Neural Networks (ANNs) [25], and Naive Bayes (NB) [10]. DL has shown a lot of success in various fields such as aerospace and defence [16], natural language processing [12], image recognition [15], speech recognition [26], medical research [27], etc. DL-based algorithms deal with large data sets, often with a multitude of features (inputs), opposite compared to traditional ML algorithms besides that deep learning is much faster than machine learning when dealing with a big size of data set. Many features are important and essential to solve a specific classification problem, some are redundant and not necessary. Also, data sets with high vector features are often challenging for training and testing. Deep Learning (DL) is our focus in this article. Deep Learning is part of Machine Learning that has to do with the structure, which is inspired by the biological neuron's operation and working function within the human brain [28]. we evaluate three models of deep learning. We build deep artificial neural networks, deep neural networks, and recurrent neural networks. For this research, we use the UNSW-NB15 Dataset for our deep learning models and evaluate the performance for each model results performed on the UNSW-NB15 dataset.

3. Materials and Methodology

This section has shown the required techniques/methods with methodological steps to fulfil this research.

3.1. Dataset description

The UNSW-NB15 data sets ' raw network packets are configured to generate hybrid real current normal activities and simulated contemporary attack behaviour using the IXIA Perfect-storm method in the cyber range Lab at the Austrian Centre for Cyber Security (ACCS). The tcpdump tool is used to archive 100 GB (e.g., Pcap files) of raw traffic. The Argus, Bro-IDS tools are utilized, and twelve algorithms are developed to generate a total of 49 features with the class label. The original dataset consists of 2,540,044 packets, which are stored in the four CSV files [29], most researchers have used these datasets to evaluate their developed intrusion detection system separately.

3.2. Proposed deep learning models

The structure and depth of the human brain influenced deep learning. The network learns to map the input function to the output due to its multiple layers of

abstraction. The learning process does not rely on the abilities of features selection engineering. According to a set of criteria, a sequence of statistical techniques may be used to determine whether a classification is correct based on the probability of error. In the area of deep learning, we concentrate on deep networks with classification training in hierarchical networks of unsupervised learning of multiple layers. The deep intrusion detection mechanisms of the network can be classified depending on the implementation of architecture and technologies. The models used for analysis are described in this section. A deep neural network classifier is the first model. The second is the convolution neural network model and the third is the long short-term memory with the recurrent neural network. They use accuracy and loss in every deeper learning and machine learning algorithm to calculate the output of these models.

3.2.1. Artificial Neural Network (ANN)

The main aspect is artificial neural networks (ANNs). Since the 1940s, ANNs have been around and used in various applications [30, 31]. A mixture of enhanced theory beginning with deep belief nets and unsupervised pre-trainings, with stronger hardware capabilities such as graphics processing units (GPUs) are the achievements of deep learning in the past decade. See for example [31-33]. Deep ANNs In fields such as image analysis, pattern recognition, target identification, natural language processing, and self-driving vehicles, the deep ANNs are regularly used with promising results to name a few fields. There are also several unresolved issues as to how and why deep ANN function so well in many critical areas of operation produced spectacular results. Since the 1990s, it has been understood from the function approximation theory that ANNs are a universal approximation to allow any continuous function and derivatives to be approximate [34-36]. Figure 1 shows the Deep ANN architecture

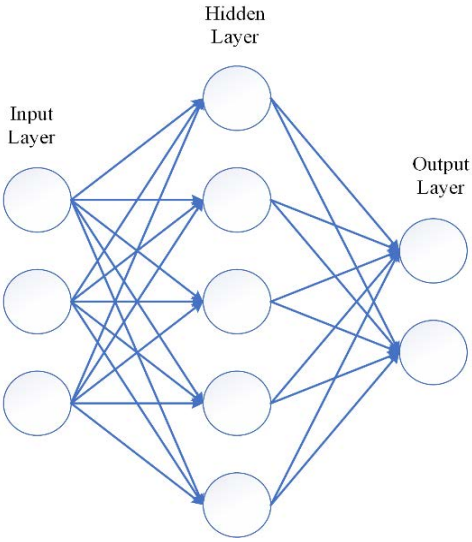


Fig. 1. Deep ANN architecture.

The ANN model hyperparameters will be described in Table 1.

Table 1. ANN hyperparameters.

Hyperparameter	Value/Type
Hidden Layer	1
Neurons	850
Optimizer	Adam
Hidden Layer Activation	Relu
Output Layer Activation	Softmax
Epochs	100
Batch size	100

3.2.2. Deep Neural Network (DNN)

A deep neural network is an artificial neural network (ANN) with several hidden layers between the input layer and the output layer is a deep neural network (DNN). The DNN seeks the right mathematical method, whether it be a linear or a non-linear relationship, to transform the input to the output. The network traverses the layers that measure each production likelihood. For instance, a DNN trained to recognize dog breeds will go over the given image and estimates the likelihood that the dog is a certain race in the picture [37].

DNNs are typically feedforward networks where data flows from the input to the output layer without a loopback, as shown in Fig. 2. The DNN generates the virtual neuron map and assigns arbitrary numerical values or "weights" to connections between neurons. The input and weights are multiplied, and the value returns between 1 and 0. An algorithm will change weights if the network did not correctly identify a sequence. This allows the algorithm to manipulate those parameters until the appropriate mathematical manipulation is decided to complete the processing of data [5]

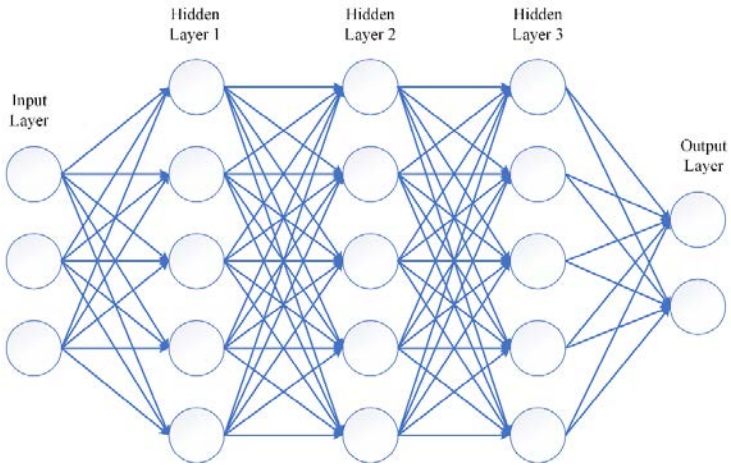


Fig. 2. DNN architecture.

The DNN model hyperparameters will be described in Table 2.

Table 2. DNN hyperparameters.

Hyperparameter	Value/Type
Hidden Layers	3
Neurons	100
Optimizer	Adam
Hidden Layer Activation	Relu
Output Layer Activation	Softmax
epochs	100
Batch size	100

3.2.3. Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM)

Recurrent neural networks, the strongest and recognized of which are LSTM's ("long short-term memory"), reflect a kind of artificial neural network that recognizes trends in sequences of the data (including, though not limited to, genomes, handwritten and spoken words) including digital time-series data from sensors, text, stock market, and government authorities [38]. What distinguishes RNNs and LSTMs from other neural networks is that they take time and sequence are considered, RNNs and LSTMs have got a temporal component. Investigations have shown that they are one of the most efficient and versatile forms of the neural network, although the focus function has recently been surpassed by memory networks, transformers, and language tasks. RNNs may also be extended to photos that can be partitioned into a series of patches and treated as fragments. We have repeated analogies with brain memory because recurrent networks provide a certain form of memory and even memory becomes part of the human experience [39, 40]. Figure 3 illustrates RNN architecture.

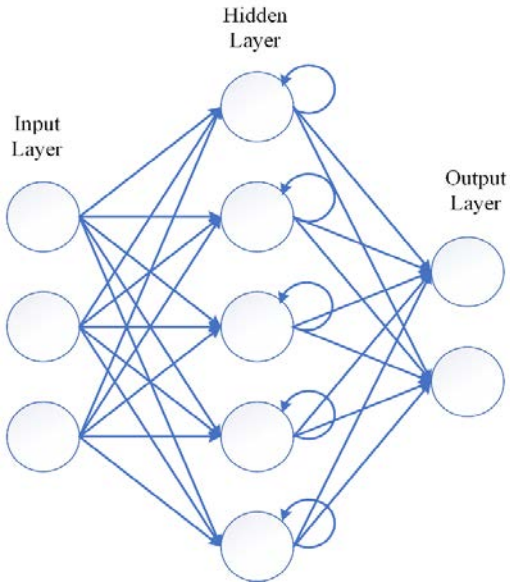


Fig. 3. RNN architecture.

The RNN model hyperparameters will be described in Table 3.

Table 3. RNN hyperparameters.

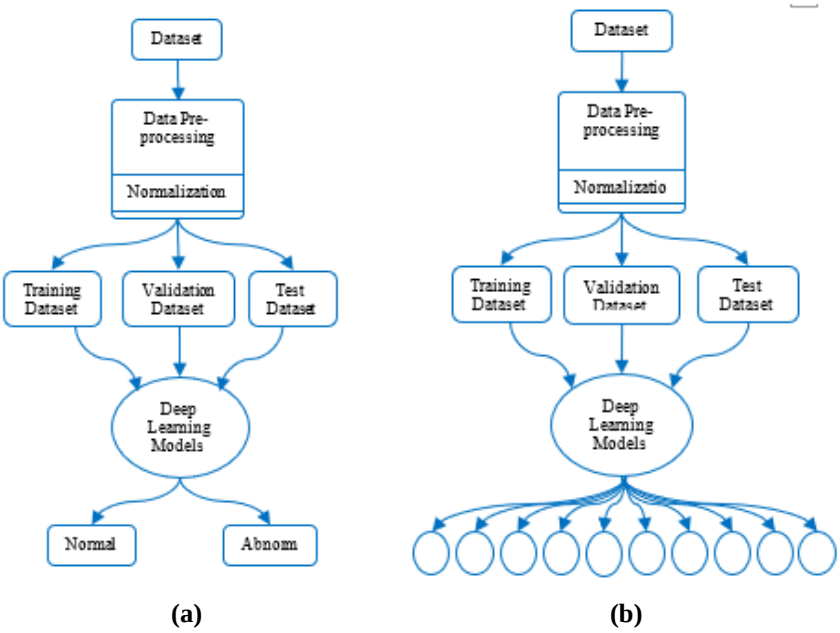
Hyperparameter	Value/Type
layers	3
Neurons	128,64,32
Optimizer	Adam
Hidden Layer Activation	Relu
Output Layer Activation	Softmax
epochs	100
Batch size	100

3.3. Proposed deep-IDS flowchart

The proposed Deep learning classifier models as the NIDS was shown in Fig. 4. Initially, the first phase (Pre-process dataset) Apply data cleaning to the input UNSW-NB15 dataset. In the second phase (Normalization) this phase will normalize UNSW-NB15 dataset by using the min-max technique. The third phase (Splitting) Split the Dataset into training, validation, and test sets. The fourth phase (Deep Learning) Construct the (DANN, DNN, and RNN) classifier with enhanced the dataset from phase two and Training the deep learning models as classifiers in binary and multi-class modes. The last phase (Evaluation) Evaluate our proposed intrusion detection system based on deep learning techniques using accuracy and loss for each model and for both classifications binary and multi-class. The detailed description of each step in the proposed phase is explained below, where the flow chart for our methodology will be in Fig. 4:

- Phase 1 (Data cleaning): This stage will be divided into two steps because the first step is that the original dataset consists of many missing values in some feature columns which cannot be fed to machine learning or specifically to deep learning models. So that in this stage we filled all empty cells with "0" value which would represent the missing value. While the second step is that converting the stored cells as text needs to be converted to numerical value where each categorical value will be represented as a specific numerical value.
- Phase 2 (Normalization): Normalization of data was especially helpful for systems in which the measurements are commonly represented on vastly different levels. Min-max normalization helps to build neural networks more consistently. This method for normalization has the advantage of accurately maintaining all data connections and therefore does not contribute to any prejudice. The increasing function is below the correct value range for the classification as min-max is added, but the respective distributions of the related features stay inside the current value range [10].
- Phase 3 (Split Dataset): In this stage, we split the main dataset into a 70% training set, 15 % for validation set, and 15 % test set.
- Phase 4 (Deep-Intrusion Detection System): we proposed an intrusion detection system based on deep learning techniques with three types of deep learning models which is (ANN, DNN, and RNN).
- Phase 5 (Evaluation): This will evaluate deep learning models as deep intrusion detection techniques and measure their performance with normal machine learning techniques based. Moreover, the authors used accuracy and loss metrics to measure the performance of each model. Furthermore, a call-back function been used toward monitoring validation loss at each epoch in

the training process means that if validation loss has not been improved for twenty epochs the training process will be interrupted, once the learning process is finished, the evaluation phase will start with the test set to evaluate the deep learning model.



(a) **(b)**

Fig. 4. Proposed deep-IDS on the improved dataset when (a) Is the binary labelling. (b) As improved multi-class labelling.

4. Results and Discussion

This section has shown the outcomes of this research.

4.1. Improved UNSW-NB15

The combined dataset consists of 1,840,046 packets which is a combination of three CSV files out of four files, we did exclude one file because the number of extracted features was less than the rest of the files and has not as been described by authors who developed the dataset. The dataset as a binary labelling and categorical labelling finds the Impact of labelling types on the classification accuracy for deep learning techniques as the intrusion detection system. Tables 4 and 5 show the dataset improved labelling details and the developed multiclass classification. We pre-process the dataset with convert the categorical features into numerical, the normalization of the dataset with min-max normalization technique, then we label the multiclass classification to be started with 0 to 9 categories.

Table 4 shows attack families and based on these families we will improve the dataset to be multi-class labelling dataset so that the improved dataset for deep learning will be again improved to be 10 classes. Furthermore, the type of features in UNSW-NB15 dataset are shown in Table 5.

Table 4. Packets and their family category.

Packets Type	Attack Family
Attack Packets	Fuzzers
	Analysis
	Backdoors
	DoS
	Exploits
	Generic
	Reconnaissance
	Shellcode
	Worms
Benign Packets	Normal

Table 5. Features description and their data type.

Name	Type	Name	Type	Name	Type
srcip	nominal	Djit	Float	Dpkts	integer
sport	integer	Stime	Timestamp	swin	integer
dstip	nominal	Ltime	Timestamp	dwin	integer
dsport	integer	Sintpkt	Float	stcpb	integer
proto	nominal	Dintpkt	Float	dtcpb	integer
state	nominal	Tcprtt	Float	smeansz	integer
dur	Float	Synack	Float	dmeansz	integer
sbytes	Integer	Ackdat	Float	ct_src_dport_ltm	integer
dbytes	Integer	is_sm_ips_ports	Binary	ct_dst_sport_ltm	integer
sttl	Integer	ct_state_ttl	Integer	ct_dst_src_ltm	integer
dttl	Integer	ct_flw_http_mthd	Integer	trans_depth	integer
sloss	Integer	is_ftp_login	Binary	res_bdy_len	integer
dloss	Integer	ct_ftp_cmd	integer	Sjit	Float
service	nominal	ct_srv_src	integer	attack_cat	nominal
Sload	Float	ct_srv_dst	integer		
Dload	Float	ct_dst_ltm	integer		
Spkts	integer	ct_src_ ltm	integer		

The dataset will be used to train Deep Learning models entrance throw input layer where it will be 48 features and then the dataset will be split into three sets as training, validation, and test with 70%, 15% and 15% in series, which will be fed to deep learning models.

4.2. Deep learning models experiment results

I t is important to notice that the flowchart in the in Fig. 4 as a general for all types of deep learning models and both classification for binary classification as in Part (a) and multi-classification as in part (b) where the pre-processed dataset will be separated into three types, which will be 70% of the original dataset as training set will be used to train deep learning model, and 15% will be as test set which will be used to evaluate/test our trained deep learning model after being trained with the training set and lastly will be separated into a validation set with 15% size which is used to validated the accuracy and loss after each training epoch to notice if there is underfitting/overfitting during the of the dataset. Notably, dataset split done randomly. Evaluation metrics were Accuracy, Loss, and training time taken by one

epoch. For average of training accuracy, average of validation accuracy, average of training loss, and average of validation loss are used to show that there is no overfitting/underfitting in the dataset. Also, total training time is the time taken to finish the training process for deep learning model, although the number of epochs when the call-back is the number of epoch complete training when call-back function stopped the training process, by dividing training time on the number of epochs completed the training we will get the training time taken for each epoch and which will be used as metric in our evaluation.

4.2.1. Artificial Neural Network (ANN)

The results of our proposed ANN model for both classifications shown in Table 6. The results of the artificial neural network with the enhanced dataset are shown in Table 6 contains results for both classifications (binary and Multi-class). The table showed the deployment of artificial neural network results for binary labelling and Multi-class labelling, in terms of accuracy, loss, and training time for each epoch. Remarkably, the artificial neural network performance in binary classification was better than Multi-class classification. The accuracy for the test set was 99.26% for the binary classification and 97.89% for the multi-class classification accuracy. While the loss of binary classification was 1.51% compared to 5.27% in multi-class classification. Results of average of training accuracy, average. of validation accuracy, average of training loss and average of validation loss are used to show that our deep learning model and the enhanced dataset are not having an over/under fitting. As it was generated in each epoch in the training process so that we took the average of 100 epoch results.

Table 6. ANN results for both classifications.

Metrics	Binary Classification	Multi-class Classification
Test Accuracy	99.26 %	97.89 %
Avg. of Training Accuracy	99.28 %	97.89 %
Avg. of Validation Accuracy	99.27 %	97.85 %
Test Loss	1.51 %	5.27 %
Avg. of Training Loss	1.42 %	5.18 %
Avg. of Validation Loss	1.46 %	5.35 %

4.2.2. Deep Neural Network (DNN)

The results of our proposed DNN model for both classifications are shown in Table 7. Deep neural network results with the enhanced dataset are shown in Table 7, which consist of result for both classifications (binary and Multi-class). the deployment of deep neural networks in the improved dataset for binary labelling and Multi-class labelling, the results in terms of accuracy, loss, and training time for each epoch. Accordingly, deep neural network performance in binary classification was better than the performance of multi-class classification.

The test set accuracy for the binary classification was 99.22% while the accuracy in multi-class classification was higher 99.59 %, and the loss in binary classification was 1.56% compared to 0.92 % in multi-class classification which is

much less than binary classification. Results of average of training accuracy, average of validation accuracy, average of training loss and average of validation loss are used to show that our deep learning model and the enhanced dataset are not having an over/under fitting. As it was generated in each epoch in the training process so that we took the average of 100 epoch results.

Table 7. DNN results for both classifications.

Metrics	Binary Classification	Multi-class Classification
Test Accuracy	99.22 %	99.59 %
Avg. of Training Accuracy	99.20 %	99.58 %
Avg. of Validation Accuracy	99.23 %	99.59 %
Test Loss	1.53 %	0.92 %
Avg. of Training Loss	1.64 %	0.96 %
Avg. of Validation Loss	1.52 %	0.92 %

4.2.3. Recurrent neural network with LSTM

The results of our proposed RNN-LSTM model for both classifications are shown Table 8. Recurrent neural network results with the enhanced dataset are shown in Table 8 which consist of result for both classifications (binary and Multi-class). the deployment of Recurrent neural networks in the improved dataset for binary labelling and Multi-class labelling, the results in terms of accuracy, loss, and training time for each epoch. Accordingly, recurrent neural network performance in binary classification was better than the performance of multi-class classification.

The test set accuracy for the binary classification was 85.42% while the accuracy in multi-class classification 85.38%, and the loss in binary classification was 35.18% compared to 48.56% in multi-class classification. Results of average of training accuracy, average of validation accuracy, average of training loss and average of validation loss are used to show that our deep learning model and the enhanced dataset are not having an over/under fitting. As it was generated in each epoch in the training process so that we took the average of 100 epoch results.

Table 8. RNN-LSTM results for both classifications.

Metrics	Binary Classification	Multi-class Classification
Test Accuracy	85.42 %	85.38 %
Avg. of Training Accuracy	85.40 %	85.40 %
Avg. of Validation Accuracy	85.43 %	85.43 %
Test Loss	35.18 %	48.56 %
Avg. of Training Loss	49.86 %	49.86 %
Avg. of Validation Loss	48.62 %	48.62 %

4.3. Discussion

This section will compare and discuss the accuracy that has been achieved in this research with the accuracy been achieved earlier with UNSW-NB15 dataset, moreover a comparison been done for both classification types as shown in Table 9. In this table, we used the accuracy produced in both labelled classifications, and

then compare these accuracies with 12 different types of machine learning and deep learning models used with UNSW-NB15 dataset listed in Table 9. Our proposed deep learning techniques are the first three highlighted rows in the table which represented our proposed techniques (ANN, DNN and RNN-LSTM) performances on UNSW-NB15 dataset, The accuracy of binary classification in the proposed ANN technique was 99.26% which is the highest accuracy achieved among all of our proposed models and related works, Also our proposed DNN technique achieved 99.22%, which is the second highest accuracy achieved among the related works and developed models. Finally, regarding accuracy of binary classification our third developed model achieved an accuracy of 85.42%, which is considered low compared to the other developed models.

Table 9. Performance comparison of our proposed techniques with related works.

Machine Learning type	Acc. In Binary	Acc. In Multi-class
Proposed ANN	99.26 %	97.89 %
Proposed DNN	99.22 %	99.59 %
Proposed RNN-LSTM	85.42 %	85.38 %
Decision tree [41]	86.13 %	78.73 %
Artificial Neural Networks [41]	86.31 %	78.14 %
Reduced Error Pruning Tree [41]	87.80 %	79.20 %
RandomTree [41]	86.59 %	76.21 %
Naïve Bias Tree [41]	80.04 %	73.86 %
Logistic Regression [42]	NAN	83.15 %
Nave Bayes [42]	NAN	82.07 %
EM Clustering [42]	NAN	78.47 %
Ramp-KSVCR [43]	NAN	93.52 %
PSI-NetVisor [44]	94.54 %	NAN
Deep ANN [45]	98.99 %	NAN
Deep belief network [46]	NAN	86.49 %

For the accuracy of multi-class classification, our proposed DNN technique recorded the highest accuracy with 99.59% versus the works in the table. then the second highest accuracy within the multi-class classification recorded for our proposed ANN technique with 97.89%. Lastly the third developed deep learning model RNN-LSTM achieved 85.38%, which is considered low when compared to other models and it comes the fifth highest accuracy among all works in the table within the accuracy of multi-class classification. Remarkably, the result of the proposed ANN and DNN recorded the highest accuracies among all other types of techniques and showed the ability of these two models to be developed as IDS. The next five different machine learning techniques presented in the table within [41], the techniques (Decision tree, Artificial Neural Networks, Reduced Error Pruning Tree, RandomTree, and Naïve Bias Tree) with (86.13 %, 86.31 %, 87.80 %, 86.59 %, and 80.04 %) accuracies respectively for the binary classification, Whereas the accuracy results for multi-class classification (78.73 %, 78.14 %, 79.20 %, 76.21 %, and 73.86 %) respectively. Furthermore, the results in [42] that used multi-class classification for machine learning types (Logistic Regression, Nave Bayes, and EM Clustering) were (83.15 %, 82.07 %, and 78.47 %) respectively, notably that the authors have not tested their models with binary classification. The accuracy in [43] for multi-class classifications was 93.52%, which was achieved using developed model called Ramp-KSVCR. the model achieved higher accuracy

compared to other machine learning techniques in the table but still less than the accuracy achieved in our two developed proposed deep learning models.

According to Mishra et al. [44] achieved an accuracy of 94.54% with PSI-NetVisor for their proposed approach in binary classification. Additionally, in [45] Al-Zewairi et al. came out with an accuracy of 98.99% with Deep ANN for binary classification, where the authors have not tested their model with multi-class classification. Finally, Tian et al. [46] proposed an improved deep belief network model to make the network fits the training data and improve performance in intrusion detection, they used to test their improved model using UNSW-NB15 dataset with categorical dataset and achieved 86.49%. The limitation of this research was due to the limited hardware capability because of which we were not able to increase the number of hidden layers and neurons to more than we have previously specified for each model, additionally hyperparameters could be investigated more to find the optimal deep learning model for this type of dataset Accordingly. Our proposed approach achieved higher accuracy in terms of binary and multi-class classification in the proposed ANN and DNN techniques, while RNN classification results were not promising and did not keep up with the related works and this leads to a conclusion that RNN is not good as an Intrusion detection system compared to ANN and DNN.

5. Conclusion

This research proposes deep learning models based on ANN, DNN, and RNN as an intrusion detection system, which is a new collaborative intrusion detection system for detecting intrusive activities in computing environments. This research comprises of the pre-processing UNSW-NB15 dataset enhanced the dataset in order to be handled with deep learning models for identifying abnormal patterns. Moreover, the deep learning architecture which does not require any of the features engineering technologies such as (e.g., feature selection methods…etc). The experimental results of proposed deep learning models show its superiority for detecting abnormal events using the improved UNSW-NB15 dataset compared with earlier techniques that have been developed on the same dataset. In the future, we plan to extend this study to deploy the framework in a real environment with further findings and explanations.

References

1. Evans, D. (2011). "The internet of things: How the next evolution of the internet is changing everything." *CISCO white paper*, 1(2011), 1-11.
2. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; and Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50, 102419.
3. Aleesa, A.M.; and Hassan, R. (2016). A proposed technique to detect DDoS attack on IPv6 web applications. *Fourth International Conference on Parallel, Distributed and Grid Computing (PDGC)*. Waknaghat, India, 118-121.
4. Gers, F. (2001). *Long short-term memory in recurrent neural networks*. Ph.D. Thesis. Computer Department Federal Institute of Technology, Lausanne, Switzerland.
5. Aleesa, A.; Zaidan, B.; Zaidan, A.; and Sahar, N.M. (2020). Review of intrusion detection systems based on deep learning techniques: coherent taxonomy,

challenges, motivations, recommendations, substantial analysis and future directions. *Neural Computing and Applications*, 32(14), 9827-9858.

6. Mbarek, B.; Ge, M.; and Pitner, T. (2020). Enhanced network intrusion detection system protocol for internet of things. *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. Brno, Czech Republic, 1156-1163.
7. Saeedi, K. (2019). *Machine learning for DDOS detection in packet core network for IoT*. Master Thesis. Luleå University of Technology, Department of Computer Science, Electrical and Space Engineering, Skelleftea, Sweden.
8. Avital, A.Z.N.; Azaria, J.; and Lambert, K. (2020). Global DDoS Threat Landscape Report. Research Laboratory: Imperva.
9. Soekhoe, D.; Putten, P.V.D., and Plaat, A, (2016). On the impact of data set size in transfer learning using deep neural networks. *International Symposium on Intelligent Data Analysis: 2016*. Stockholm, Sweden, 50-60.
10. Myint, H.O.; and Meesad, P. (2009). Incremental learning algorithm based on support vector machine with Mahalanobis distance (ISVMM) for intrusion prevention. *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. Pattaya, Chonburi, 630-633.
11. Farnaaz, N.; and Jabbar, M.J. (2016). Random forest modeling for network intrusion detection system. *Procedia Computer Science*, 89(1), 213-217.
12. Al-Qatf, M.; Lasheng, Y.; Al-Habib, M.; and Al-Sabahi, K.J.I.A. (2018). Deep learning approach combining sparse autoencoder with SVM for network intrusion detection. *IEEE Access*, 6, 52843-52856.
13. Peddabachigari, S.; Abraham, A.; and Thomas, J. (2004). Intrusion detection systems using decision trees and support vector machines. *International Journal of Applied Science and Computations*, 11(3), 118-134.
14. Chowdhuri, S.; Das, S.K.; Roy, P.; Chakraborty, S.; Maji, M.; and Dey, N. (2014). Implementation of a new packet broadcasting algorithm for MIMO equipped Mobile ad-hoc network. *International Conference on Circuits, Communication, Control and Computing*. Bangalore, India, 372-376.
15. Chandre, P.R.; Mahalle, P.N.; Shinde, G.R. (2020). Deep learning and machine learning techniques for intrusion detection and prevention in wireless sensor. *Design Frameworks for Wireless Networks*, 82, 95.
16. Muna, A.H.; Moustafa, N.; Sitnikova, E. (2018). Identification of malicious activities in industrial internet of things based on deep learning models. *Journal of Information Security and Applications*, 41, 1-11.
17. Shi, H.; Li, H.; Zhang, D.; Cheng, C.; Cao, X. (2018). An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification. *Computer* Network, 132, 81-98.
18. Fong, S.; Li, J.; Song, W.; Tian, Y.; Wong, R.K.; and Dey, N. (2018). Predicting unusual energy consumption events from smart home sensor network by data stream mining with misclassified recall. *Journal of Ambient Intelligence and Humanized Computing*, 9(4), 1197-1221.
19. Mukherjee, A.; Keshary, V.; Pandya, K.; Dey, N.; and Satapathy, S.C. (2018). Flying ad hoc networks: A comprehensive survey. *Information and Decision Sciences*. Springer, Singapore, 569-580.

20. Qi, Y. (2012). Random forest for bioinformatics. *Ensemble machine learning.* Springer, Singapore, 307-323.
21. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; and Greer, K. (2003). KNN model-based approach in classification. *OTM Confederated International Conferences on the Move to Meaningful Internet Systems*. Berlin, Heildeberg, 986-996.
22. Cheong, S.; Oh, S.H.; and Lee, S.Y. (2004). Support vector machines with binary tree architecture for multi-class classification. *Neural Information Processing*, 2(3), 47-51.
23. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I; and Dorronsoro, B. (2020). Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Applied Sciences*, 10(5), 1775.
24. Diao, R.; Sun, K.; Vittal, V.; O'Keefe, R.J.; Richardson, M.R.; Bhatt, N.; Stradford, D.; and Sarawgi, S.K. (2009). Decision tree-based online voltage security assessment using PMU measurements. *IEEE Transactions on Power Systems*, 24(2), 832-839.
25. Priddy, K.L.; and Keller, P.E. (2005). *Artificial neural networks: an introduction*. Bellingham, USA: SPIE Press.
26. Noda, K.; Yamaguchi, Y.; Nakadai, K.; Okuno, H.G.; Ogata, T. (2015). Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4), 722-737.
27. Shen, D.; Wu, G.; and Suk, H.-II (2017). Deep learning in medical image analysis. *Annual Review of Biomedical* Engineering, 19, 221-248.
28. Arel, I.; Rose, D.C.; and Karnowski, T.P. (2010). Deep machine learning-a new frontier in artificial intelligence research (research frontier). *IEEE Computational Intelligence Magazine*, 5(4), 13-18.
29. Moustafa, N.; and Slay, J. (2015). A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 Military Communications and Information Systems Conference (MilCIS)*. Canberra, ACT, 1-6.
30. McCulloch, W.S.; and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115-133.
31. Safara, F.; Souri, A.; and Serrizadeh, M. (2020). Improved intrusion detection method for communication networks using association rule mining and artificial neural networks. *IET Communications*, 14(7), 1192-1197.
32. Hinton, G.E.; Osindero, S.; and Teh, Y-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7), 1527-1554.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1, 1097-1105.
34. Hornik, K.; Stinchcombe, M.; and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Network*, 2(5), 359-366.
35. Li, X. (1996). Simultaneous approximations of multivariate functions and their derivatives by neural networks with one hidden layer. *Neurocomputing*, 12(4), 327-343.

36. Hornik, K.; Stinchcombe, M.; and White, H. (1990). Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5), 551-560.
37. Kwon, O.; Kim, H.G.; Ham, M.J.; Kim, W.; Kim, G.-H.; Cho, J.-H.; Kim, N.I.; and Kim, K. (2020). A deep neural network for classification of melt-pool images in metal additive manufacturing. *Journal of Intelligent Manufacturing*, 31(2), 375-386.
38. Muhuri, P.S.; Chatterjee, P.; Yuan, X.; Roy, K.; and Esterline, A. (2020). Using a long short-term memory recurrent neural network (LSTM-RNN) to classify network attacks. *Information*, 11(5), 243.
39. Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.J.; and Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *Computer Vision and Pattern Recognition*, 2, arXiv:1502.04623
40. Almiani, M.; AbuGhazleh, A.; Al-Rahayfeh, A.; Atiewi, S.; and Razaque, A. (2020). Deep recurrent neural network for IoT intrusion detection system. *Simulation Modelling Practice and Theory*, 101, 102031.
41. Belouch, M.; Hadaj, S.E.; and Idhammad, M. (2017). A two-stage classifier approach using RepTree algorithm for network intrusion detection. *International Journal of Advanced Computer Science and Applications*, 8(6), 389-394.
42. Moustafa, N.; and Slay, J. (2016). The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. *Information Security Journal: A Global Perspective*, 25(1-3), 18-31.
43. Bamakan, S.M.H.; Wang, H.; and Shi, Y. (2017). Ramp loss K-Support Vector Classification-Regression: A robust and sparse multi-class approach to the intrusion detection problem. *Knowledge-Based Systems*, 126, 113-126.
44. Mishra, P.; Pilli, E.S.; Varadharajan, V.; and Tupakula, U. (2017). PSI-NetVisor: Program semantic aware intrusion detection at network and hypervisor layer in cloud. *Journal of Intelligent and Fuzzy* Systems, 32(4), 2909-2921.
45. Al-Zewairi, M.; Almajali, S.; and Awajan, A. (2017). Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system. *2017 International Conference on New Trends in Computing Sciences (ICTCS)*. Amman, Jordan, 167-172.
46. Tian, Q.; Han, D.; Li, K.-C.; Liu, X.; Duan, L.; and Castiglione, A. (2020). An intrusion detection approach based on improved deep belief network. *Applied Intelligence*, 10, 3162-3178.