Scientific African 9 (2020) e00497

Contents lists available at ScienceDirect

Scientific African

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

Network intrusion detection system using supervised learning paradigm

J. Olamantanmi Mebawondu[a,*], Olufunso D. Alowolodu[b], Jacob O. Mebawondu[a], Adebayo O. Adetunmbi[c]

[a] Department of Computer Science, Federal Polytechnic, Nasarawa, Nigeria
[b] Department of CyberSecurity, Federal University of Technology, Akure, Nigeria
[c] Department of Computer Science, Federal University of Technology, Akure, Nigeria

ARTICLE INFO

Article history:
Received 17 February 2020
Revised 20 July 2020
Accepted 24 July 2020

Keywords:
Artificial neural network
Multi-layer perceptron
Gain ratio
Accuracy
UNSW-NB15 dataset

ABSTRACT

Internet has positively changed social, political and economic structures and in many ways obviating geographical boundaries. The enormous contributions of Internet to business transactions coupled with its ease of use has resulted in increased number of internet users and consequently, intruders. It is crucial to safeguard computer resources with the aid of Intrusion Detection Systems (IDS) in addition to Intrusion Prevention Systems. In recent times, enormous network traffic generated in terabytes within couples of seconds are difficult to analyze with the traditional rule-based approach; hence, researchers have to subject data mining techniques to intrusion detection with emphasis on intrusion detection accuracy; relevant feature selection leads to faster and enhanced accurate detection rate. Therefore, this paper presents a light weight IDS based on information gain and Multi-layer perceptron Neural Network. Gain ratio was used in selecting relevant features for attack and normal traffic prior classification using Neural Network. Empirical results from the UNSW-NB15 intrusion detection dataset on thirty selected attributes is a highly ranked decision, thus, the light weight IDS is suitable for real time intrusion detection.

Introduction

The internet has completely revolutionized the world in business transactions, source of information, communication and socialization, among others. It has contributed immensely to nations' economic growth and has greatly impacted the numerical growth of computer network users [10]. However, the internet usability increase is inherently bogged with information systems security which increases in sophistication daily [2,18].

The computer network technology is advancing and changing every day with its increasing popularity. Topology, media, protocol, and security are vital components of computer network; the biggest concern, however, has been security. The

* Corresponding author at: P. O. Box 169, Nasarawa, LGC, Nasarawa, Nigeria
E-mail addresses: mebawonduojosphine@fedpoly.nas.edu (J.O. Mebawondu), odalowolodu@futa.edu.ng (O.D. Alowolodu), aoadetunmbi@futa.edu.ng (A.O. Adetunmbi).

https://doi.org/10.1016/j.sciaf.2020.e00497

network and its security are of paramount importance in the present data communication environment. On the other hand, whole security breaches blockage appears at present unrealistic, but intrusion attempts could be detected so that action may be taken to mitigate attack impacts on the network damage [4,11].

IDS becomes a necessity when it is clear that intrusion prevention systems such as encryption, access control and firewall, among others are not sufficient in curbing the menace of computer and network security. The critical design terminologies of IDS in terms of their efficiency and analysis are paramount issues to focus on. It is paramount when considering machine learning, as an offshoot of Artificial Intelligence (AI), in design detection technique such as Neural Network and Naïve Bayes [21].

The goal of AI via machine leaning techniques relative to the work is to build an intelligent IDS model for computer network security as a second line of defense. This intrusion detection is a process of overseeing the events happening in a computer system or network and analyzing them for marks of encroachment. It is essential to strive for full level protection that ensures safe and trusted information in this era of Information and Communication Technology (ICT). Therefore, computer networking is evolved with the development of new protocols, applications in securing and promoting its use in order to accommodate more users; also, there is an increase in the number of attacks and sophistication. All these have resulted in changing and increasing the network traffic. This necessitates the need for constant update of intrusion detection models based on machine learning algorithms for effectiveness premised on current realities of network traffic.

In the last three decades, data mining and machine learning techniques have been subjected to extensive research in developing IDS using different intrusion detection datasets, most prominent among them are the KDD99 and NSL-KDD intrusion detection datasets. Several machine learning techniques used on these datasets are rough sets, k-Nearest Neighbor [2,18], Bayes, Neural Network, Decision tree and k-means [5,18,21].

This study develops an intrusion detection model using information gain feature selection technique and artificial neural network (ANN) on recent UNSW-NB15 intrusion detection dataset that reflects currents trends in today's network traffic. The IDS introduced in this research is an anomaly IDS which is capable of detecting known and novel attacks. The rest of the paper is organized as follows: in Section 2 is literature review followed by material and methods in Section 3. Section 4 gives the experimental setup, results and discussions with conclusion in Section 5.

Literature review

Research shows that intrusions on computer networks result into loss of trillion USD annually and often times results in an individual losing his or her life savings. Due to this adverse effect, several efforts made by various researchers in curbing this menace are either by developing a network or host IDS to detect the activities that could compromise the confidentiality, integrity, and availability of computer and network resources. Existing studies concentrated on misuse intrusion detection are predominantly rule-based. The results of such studies are characterized by poor detection rate because of inability to unknown signature [1,11]. The enormous network traffic and low detection rate necessitate the need for machine learning techniques in intrusion detection to curbing the problem of intrusions on networks. Some of the machine learning techniques used to deploy an anomaly IDS are Naïve Bayes, Neural Networks, Fuzzy Logic, k Nearest Neighbor algorithms, Bagging, Random Forest, and many more [11,14]. An anomaly IDS is much more effective in detecting novel attacks though with high false alarm rate. Previous efforts include studies conducted by Boppana and Su [4], Schmidhuber. [20] and Vinchurkar and Reshamwala [21] that used different neural variants network to distinguish between normal and attack traffics in KDD99 and NSL-KDD intrusion detection datasets.

Neethu [15] opined that machine learning approach can consist of dimensionality reduction module and classification of network traffic. The importance of feature selection in preprocessing is presented by Osanaiye et al [19] stating the use of information gain, Gain ratio, Chi-squared to select relevant features in order to improve the learning rate and reduce the false alarm rate. Onyekwelu et al [17] also stated the importance of preprocessing using various methods of data transformation and data discretization. Garg [6] proposed the use of binarization technique on data, the selection of most optimal binarization algorithm gives a different performance on different datasets. Ennert et al [5] reports IDS model using several intrusion detection tools provides a set of selected tests on the performance of the IDS model that would enlarge the functionality of IDS.

Adetunmbi [1] and Neethu [15] performed dimensionality reduction on KDD99 dataset prior developing a classification models for intrusion detection. The former used rough-set and dependency ratio to select relevant attributes for intrusion detection as the first phase that proceeds ID classification models with kNN, Roughsets and Naïve Bayes among others, while the latter employed principal component analysis for dimensionality reduction and Naïve Bayes algorithm for classification. The deployed techniques result in fast and more accurate detection rate. All these models are based on old intrusion detection datasets which may not reflect true reality of the present times. However, Guo et al [7] emphasize the need to extend previous researches to include more novel attacks.

In the light of the above research gap, this paper develops a lightweight network IDS based on ANN using the recent UNSW-NB15 dataset as a benchmark for computer network security. IDS will address ICT security challenges in Africa, which will address the goals 8, 13, and 19 of African Agenda 2063 goal and the UN SDGs [12].

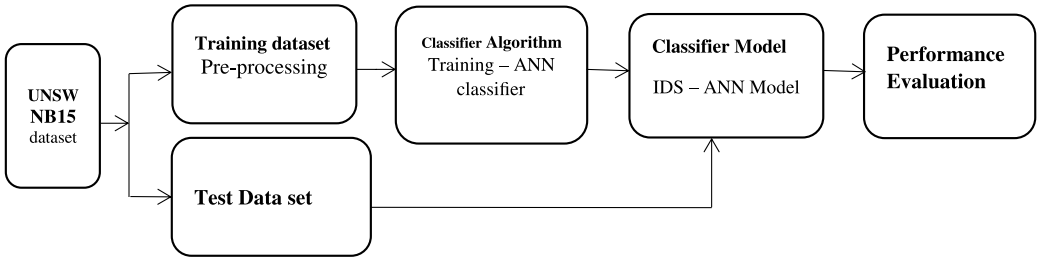


Fig. 1. System work flow.

Material and methods

This section describes the methodology employed in developing ANN based IDS. This section is divided into three sub-sections: Dataset Description, Preprocessing and Classification. This paper adopted the same approach used in the studies of Zhai and Chena, [22] and AlSallal, [3]. The experimental stages are preprocessing of the captured dataset, classification process, training and testing using required data, and finally, implementation evaluation. The training dataset is fed to the preprocessing system; the output from the preprocessing stage is sent to the classifier model. Thereafter, the test dataset is fed into the IDS system and the output is tested for performance. Fig. 1 depicts the system workflow of this study.

Fig. 1 describes the steps involved in developing the NIDS system using ANN as a classifier. In order to address the system workflow, the experimental measurements needed in this study adopted existing techniques using binarization and Gain ratio feature ranking at the dataset preprocessing stage. The subsequent steps involve training and testing using the classifier. Lastly, the performance evaluation is evaluated using the standard metrics such as accuracy, and Matthew Correlation coefficient (MCC) among others.

UNSW-NB15 dataset

UNSW-NB15 intrusion detection dataset was generated under synthetic environment at the University of New South Wales (UNSW) cybersecurity laboratory in 2015. This dataset was generated as a result of the flaws in the existing datasets such as KDD-98, KDD-CUP99 and NSL-KDD [8,16] which are characterized by defects of unavailability of a comprehensive network based attributes that can reflect current network traffic sensitivities, vast varieties of low footprint intrusions and depth structured information about network traffic.

The UNSW-NB15 dataset, uniquely captured and developed for training an IDS, is used in this study as the benchmark dataset, consisting of 45 features and 9 types of attacks for 82,332 records for the training set. The UNSW-NB15 intrusion detection dataset created by Cyber Range Lab of the Australian Centre for Cyber Security (ACCS) is sufficient for the study requirement [13].

The 9 types of attacks contained in the UNSW-NB15 dataset namely, Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms, contains 45 distinct features. These features can be nominal and non-nominal values such as float, binary, timestamp and integer.

The features are grouped into Flow, Basic, Content, Time features and, Labelled features as stated in Table 1 and in Fig. 2 give the description of the dataset features:

i Flow features contains transaction protocol, source Ip address, etc.
ii Basic features are state, total duration read, service etc.
iii Content features contains source TCP window advertisement, destination TCP etc.
iv Time feature are jitter, destination jitter, record start-time and record last time etc.
v Then the labelled features are the attack_cat and label.
vi The rest features contains additional features which are derived from matched features.

Data preprocessing

Two major tasks were carried out in this section for data reduction: normalization and feature selection. The process commences with binarization of continuous features and discretization of nominal features followed by feature ranking, a process required to remove redundant feature in the dataset using information entropy - Gain ratio. For instance, a nominal attribute "Proto" contains five attribute values - 3pc, a/N, Aes sp3 d, Any, and Argus; which are simply assigned discrete values from 0 to 4 respectively as shown in Table 2. It shows typical description of discretization of nominal attributes.

Binarization is a simple but straightforward means of discretizing data with continuous attributes using 0 and 1. It is used also in the process of assigning threshold to numerical features to obtain Boolean values, by computing average and standard derivation(s) respectively.

Table 1
Dataset features prior gain ratio features selection.

S/No.	Features	Descrp
1	id	integer
2	dur	integer
3	proto	nominal
4	service	nominal
5	state	nominal
6	Spkts	integer
7	Dpkts	integer
8	sbytes	Integer
9	dbytes	Integer
10	rate	integer
11	sttl	Integer
12	dttl	Integer
13	Sload	Float
14	Dload	Float
15	sloss	Integer
16	dloss	Integer
17	Sintpkt	Float
18	Dintpkt	Float
19	Sjit	Float
20	Djit	Float
21	swin	integer
22	stcpb	integer
23	dtcpb	integer
24	dwin	integer
25	tcprtt	Float
26	synack	Float
27	ackdat	Float
28	smean	integer
29	dmean	integer
30	trans_depth	integer
31	respose_body_len	integer
32	ct_srv_src	integer
33	ct_state_ttl	Integer
34	ct_dst_ltm	integer
35	ct_src_dport_ltm	integer
36	ct_dst_sport_ltm	integer
37	ct_dst_src_ltm	integer
38	is_ftp_login	Binary
39	ct_ftp_cmd	integer
40	ct_flw_http_mthd	integer
41	ct_src_ ltm	integer
42	ct_srv_dst	integer
43	is_sm_ips_ports	Binary
44	attack_cat	nominal
45	Label	Binary

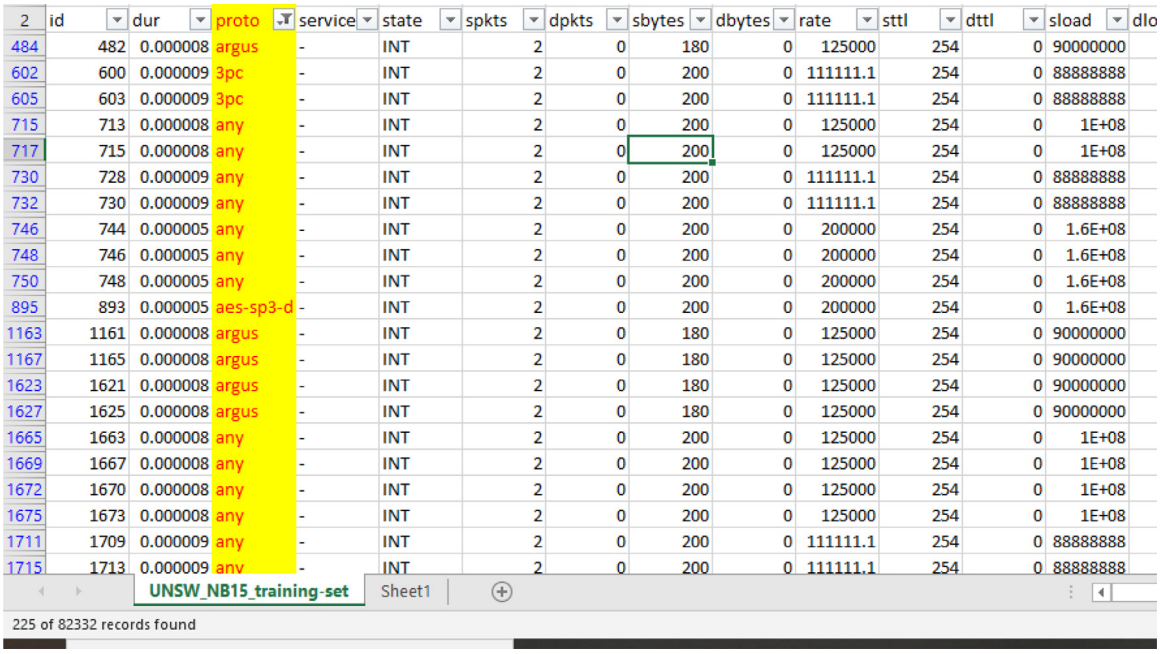
Table 2
Key of proto discretization.

S/N	Classes of proto feature	Discretized value
1	3pc	0
2	a/N	1
3	Aes sp3 d	2
4	Any	3
5	Argus	4

Binarization

The average and standard deviation value of each feature is computed for the preprocessing stage in binarization. The binarization method is used to convert the continuous variables of the dataset into a discrete value. The binarization is computed using Eqs. (1) and (2):

$$\text{Average} = \frac{\sum_{x=1}^{n} \text{ri}}{n} \tag{1}$$



2	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dlo
484	482	0.000008	argus	-	INT	2	0	180	0	125000	254	0	90000000	
602	600	0.000009	3pc	-	INT	2	0	200	0	111111.1	254	0	88888888	
605	603	0.000009	3pc	-	INT	2	0	200	0	111111.1	254	0	88888888	
715	713	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
717	715	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
730	728	0.000009	any	-	INT	2	0	200	0	111111.1	254	0	88888888	
732	730	0.000009	any	-	INT	2	0	200	0	111111.1	254	0	88888888	
746	744	0.000005	any	-	INT	2	0	200	0	200000	254	0	1.6E+08	
748	746	0.000005	any	-	INT	2	0	200	0	200000	254	0	1.6E+08	
750	748	0.000005	any	-	INT	2	0	200	0	200000	254	0	1.6E+08	
895	893	0.000005	aes-sp3-d	-	INT	2	0	200	0	200000	254	0	1.6E+08	
1163	1161	0.000008	argus	-	INT	2	0	180	0	125000	254	0	90000000	
1167	1165	0.000008	argus	-	INT	2	0	180	0	125000	254	0	90000000	
1623	1621	0.000008	argus	-	INT	2	0	180	0	125000	254	0	90000000	
1627	1625	0.000008	argus	-	INT	2	0	180	0	125000	254	0	90000000	
1665	1663	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
1669	1667	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
1672	1670	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
1675	1673	0.000008	any	-	INT	2	0	200	0	125000	254	0	1E+08	
1711	1709	0.000009	any	-	INT	2	0	200	0	111111.1	254	0	88888888	
1715	1713	0.000009	any	-	INT	2	0	200	0	111111.1	254	0	88888888	

Fig. 2. Sample set.

$$s = \sqrt{\frac{\sum (x - \bar{x})^2}{n - 1}} \tag{2}$$

where, s is the standard deviation, n is the number of items and x is the total sum of all numbers.

Gain ratio

In order to extract the most relevant features for processing, there is need to rank the features by generating ranks merits to improve supervised learning, the Gain ratio feature selection is used for ranking the attributes. Mathematically, for the feature selection, Gain ratio is computed using Eqs. (3)–(6).

$$H(S) = -\sum_{i=1}^{n} P_i Log_2 P_i \tag{3}$$

where P_i is the proportion of instances with class i; and a class is a category to which an instance may belong with a certain probability. [9].

The Gain Information formula stated in Eq. (4).

$$Gain(S, A) = H(S) - \sum_{t \in (A)} \frac{|S_t|}{|S|} H(S_t) \tag{4}$$

where t is all members of A, A is a given attribute, S is a sample of the dataset and S_t is subset where $X_A = T$

In order to avoid unnecessary biases in the result, the Eq. (5) is used to calculate the split information.

$$\text{Split info } (S, A) \;=\; -\sum_{i=1}^{r} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|} \tag{5}$$

where absolute S_i is the proportion of instances with class i within the integer range of values r.

Then, the actual Gain ratio is calculated using Eq (6) to find the ratio between the Gain information and the Split information.

$$\text{Gain Ratio} = \frac{\text{Gain info}}{\text{Split Info}} \tag{6}$$

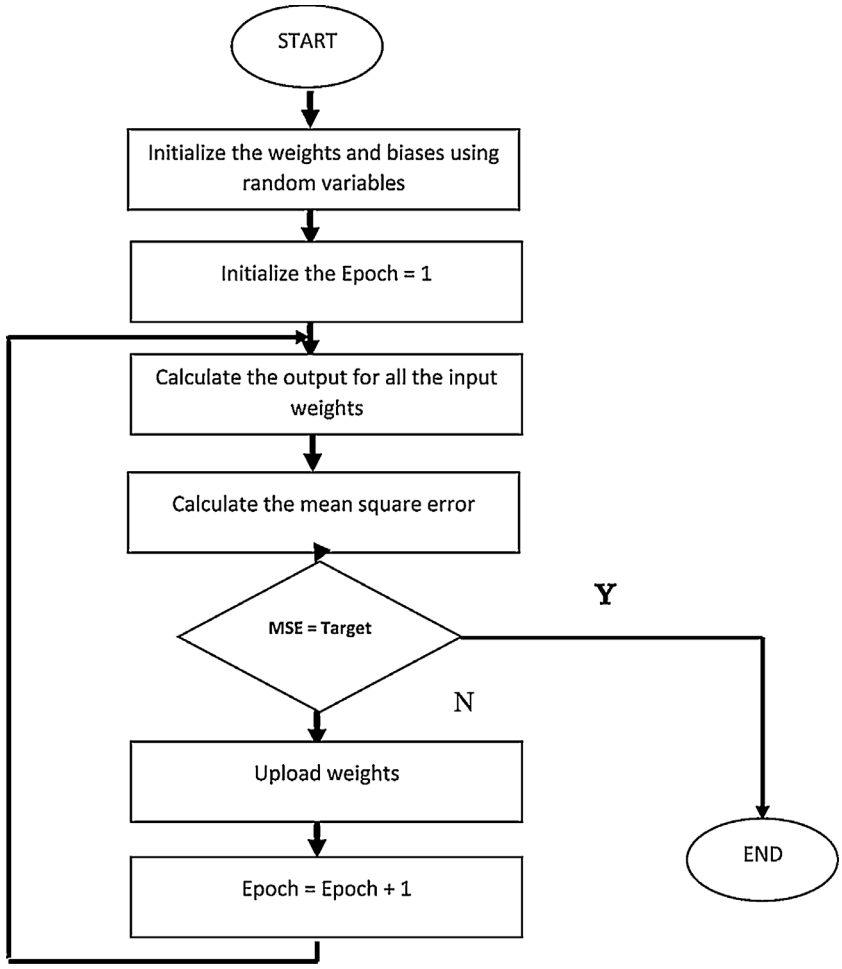


Fig. 3. Neural network system flow.

Neural network model

Dataset classification using Neural Network, the Multi-layer perceptron architecture of the ANN is applied as a classification model for IDS using Eqs (7) and (8). The formula of a Neural Network given in Eq. (7).

$$F(I) = f\left(\sum_{j} W_j .\ I_j\right) \tag{7}$$

where F(I) is predicted label, W_j is the weight of each input I_j. and f is the sigmoid activation function as shown in Eq. (8), which maps the input x to values between 0 and 1.

$$f = \frac{1}{1 + e^{-x}} \tag{8}$$

The flow of the Neural Network system begins with entering of the various input and output of the dataset. The initialization of the weights and bias randomly selected in the first instance but is computed in the next instances. The epoch is initialized as 1. The system repeatedly continues until sentinel is achieved. The sentinels are when either weights and bias hardly change alternatively like other criteria when errors are below standard or when the maximum number of iterations has reached.

The flowchart of the system is described in Fig. 3 indicating the Neural Network Process

Table 3
Dataset partitions.

Class name Data partition (%)	Yes Train (70%)	Test (30%)	No Train (70%)	Test (30%)
Number of instances	31717	13615	25908	11092
Class total	45332	37000		
Total dataset	82332			

Table 4
Undiscretized features (Extracted raw data from UNSW data).

Dur	Proto	Service	State	Spkts	Dpkts	Sbytes	Dbytes	Rate
0.000011	Udp	-	INT	2	0	496	0	90909.09
0.000008	Udp	-	INT	2	0	1762	0	125000
0.000005	Udp	-	INT	2	0	1068	0	200000
0.000006	Udp	-	INT	2	0	900	0	166666.7
0.00001	Udp	-	INT	2	0	2126	0	100000
0.000003	Udp	-	INT	2	0	784	0	333333.3
0.000006	Udp	-	INT	2	0	1960	0	166666.7
0.000028	Udp	-	INT	2	0	1384	0	35714.29
0	Arp	-	INT	1	0	46	0	0
0	Arp	-	INT	1	0	46	0	0

Performance evaluation

The performance evaluation metrics used to generate accuracy and False Alarm Rate of the classification are given in Eqs. (9) and (10)using the confusion matrix generated from the Neural Network classification result.

$$\text{Accuracy} = (TP + TN)/(TP + TN + FP + FN) \tag{9}$$

$$\text{False Alarm} = (FP + FN)/(TP + TN + FP + FN) \tag{10}$$

In the above Eqs. (9) and (10), TP means true positive, that is, attack data are detected as attack. TN means true negative, that is, normal data are not detected as normal; FP means false positive, that is, normal data are detected as attack, and FN means false negative, that is, attack data are detected as normal.

The MCC is used in machine learning as a measure of quality of binary (2-class) classification, usually used in binary classification. MCC is a correlation coefficient between the exact and predicted binary classification, usually return value of 0 or 1.

$$\text{MCC} = \frac{TP\ X\ TN - FP\ X\ FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

Experimental setup, results and discussions

For this experiment, a total of 82,332 instances of network traffic from UNSW-NB15 was used for both training and testing. Table 3 shows the training and testing data split proportion for the model formulation and testing based on the binary class label. Seventy percent (70%) for training and thirty percent (30%) for testing.

Prior to splitting, the entire dataset to be used is discretized, the nominal data were discretized based on the concept explained in Section 3.2 while discretization of the continuous attribute is achieved based on binarization. For illustration, Table 4 shows a sample raw data extracted from the UNSW-NB15 dataset. Table 5 shows the computation of the threshold based on table using binarization. The threshold is the average value computed value for each continuous feature, and a binary number 0 is assigned to the value less than the threshold and 1 otherwise.

Table 6 gives the final output after discretization. This shows typical computations perform on all the network instances available in the training set to obtain a reduced dataset prior to ANN-IDS model.

Feature selection and ANN –MLP model results

Feature ranking technique – Gain ratio discussed in Section 3.2 was used in ranking all the conditional features of UNSW-NB15 using equations in the section. The results obtained sorted in descending order is shown in Table 7; the irrelevant features were eliminated using a threshold of 0 to 0.00100 ranks.

Table 7 shows the results of the rank and merit of all the features. It shows feature ct_State_ttl has the highest ranking with the merit of 0.36882. The next feature contains ranks and their merits. The best of the ranks were determined by removing least ranks 0 to 0.00100 ranks from the list. Then, 30 features are left as the best ranks. Fig. 4 shows the graphical

Table 5
Threshold computation.

Attributes	Minimum	Average	Maximum	Standard Deviation	BINARIZATION 0	1
Dur	0	1.006756146	59.999989	4.710415804	below 1	above 1
Proto	131 unique discretized values					
Service	13 unique discretized values					
State	7 unique discretized values					
Spkts	1	18.66647233	10646	133.9155402	below 19	above 19
Dpkts	0	17.54593597	11018	115.5733837	below 18	above 18
Sbytes	24	7993.908165	14355774	171641.2195	below 7994	above 7994
Dbytes	0	13233.78556	14657531	151470.5362	below 13234	above 13234
Rate	0	82410.88674	1000000.003	148619.4645	below 82411	above 82411

Table 6
Discretized features.

Dur	Proto	Service	State	Spkts	Dpkts	Sbytes	Dbytes	Rate
0	117	-	1	0	0	0	0	1
0	117	-	1	0	0	0	0	1
0	117	-	1	0	0	0	0	1
0	117	-	1	0	0	0	0	1
0	117	-	0	0	0	0	0	1
0	117	-	0	0	0	0	0	1
0	117	-	0	0	0	0	0	1
0	117	-	0	0	0	0	0	0

representations of the merit scores of features selected using Gain ratio and depicts more clearly the degree of a feature relevance. The discretization and Gain ratio are implemented using c# programming language.

Finally, the classification model using the ANN Multi-layer perceptron. ANN MLP was then based on the selected first thirty features of the UNSW-NB15. The Matrix Laboratory (MATLAB) 8.1 is used in implementing the ANN-MLP IDS on the 30 selected attributes while the output set contained 2 features, namely: normal and attack such that if normal is 0, then the attack is 1. These sets is imported into the MATLAB workspace for the ANN training model. A total of fifty-seven thousand, six hundred and twenty-five instances (57,625) was used for training the model while twenty-four thousand, seven hundred and seven (24,707) instances were used in evaluating the performance of the model. The testing instances are made up of 11,092 and 13,615 instances of class No and Yes while No stands for absence of attack, which is, normal traffic and Yes indicates attack. Table 8 shows the confusion matrix obtained from testing evaluation.

The confusion matrix shows that out of 11,092 classified, 9,978 were correctly classified as normal, and 1,114 incorrectly classified as attack. Then, out of 13,615 classified as attack, 9,037 were correctly classified as attack, and 4,587 incorrectly classified as normal.

From the confusion matrix, TP = 9978, FP = 1114, FN = 4578 and, TN = 9037
From Eq. (9) accuracy obtained thus:
Accuracy = (9978 + 9037)/ (9978 + 9037+1114+4578) = 0.7696 = 76.**96**%
From Eq. (10),
False alarm rate= (1114+4578)/ (9978 + 9037+1114+4578) = 0.23038 = 23.**95**%

Computing MCC as a metrics (Eq. (11)) to check the balancing of the result derived from the classification, the derived data from confusion matrix of Table 8.

$$MCC = \frac{(9978\ X\ 9037) - (1114\ X\ 4578)}{\sqrt{(9978+1114)(9978+4578)(9037+1114)(9037+4578)}}$$

$$MCC = \frac{(90,171,186) - (5,099,892)}{\sqrt{(11092)(14,556)(10,151)(13615)}}$$

$$MCC = \frac{(85,071,294)}{\sqrt{\left(2.2314048941\ X\ 10^{16}\right)}}$$

$$MCC = \frac{(85,071,294)}{(149,378,877.158)}$$

$$MCC = 0.5695 \approx 0.6$$

Table 7
Gain ratio results.

GAIN RATIO		
RANK	Attribute	Merit
1	ct_state_ttl	0.36882
2	ct_dst_sport_ltm	0.27536
3	sttl	0.23069
4	Dload	0.2247
5	dttl	0.22397
6	rate	0.2083
7	state	0.18303
18	is_sm_ips_ports	0.14687
9	sload	0.13004
10	swin	0.12868
11	dwin	0.10088
12	ct_src_dport_ltm	0.10075
13	proto	0.09979
14	service	0.07073
15	ct_dst_src_ltm	0.07037
16	dpkts	0.07008
17	dbytes	0.06956
18	ct_srv_src	0.06452
19	ct_srv_dst	0.06173
20	stcpb	0.06112
21	dtcpb	0.06044
22	dmean	0.05475
23	sloss	0.05005
24	ct_src_ltm	0.04326
25	sinpkt	0.04085
26	ct_dst_ltm	0.03433
27	spkts	0.0338
28	dloss	0.03138
29	smean	0.03066
30	sjit	0.02334
31	synack	0.01632
32	response_body_len	0.01266
33	dinpkt	0.01169
34	dur	0.00723
35	sbytes	0.00696
36	ct_flw_http_mthd	0.00496
37	trans_depth	0.00494
38	ackdat	0.00395
39	is_ftp_login	0.00372
40	tcprtt	0.00349
41	ct_ftp_cmd	0.00304
42	djit	0

Table 8
Confusion matrix.

Class _IDS (number)	No	Yes
No (11,092)	9978	1114
Yes (13,615)	4578	9037

Table 9
Performance evaluation results.

Correct classification	Incorrect classification	Accuracy (%)	Classification error	TP rate	FP rate	Precision	Test time (s)
19015	5685	76.96	0.23	0.77	0.206	0.798	0.25

The performance evaluation parameters were evaluated using FP, FN, TP, and TN for validating the performance of the ANN on all the dataset. The plot of the number of epochs reached by the ANN achieved within the best performance of **76.96%** reached at epochs 200. Table 9 shows the evaluation results of the IDS model built using the Neural Network classification.

The accuracy, which is the proportion of datasets correctly classified, expressed as 76.96%, whereas 23% is the classifier error. The False Positive Rate (FPR) is the proportion of the actual number of cases incorrectly classified as **23.95%**. The Precision that is the proportion of predicted cases that are correctly classified has a value of 79.80%. The obtained accuracy

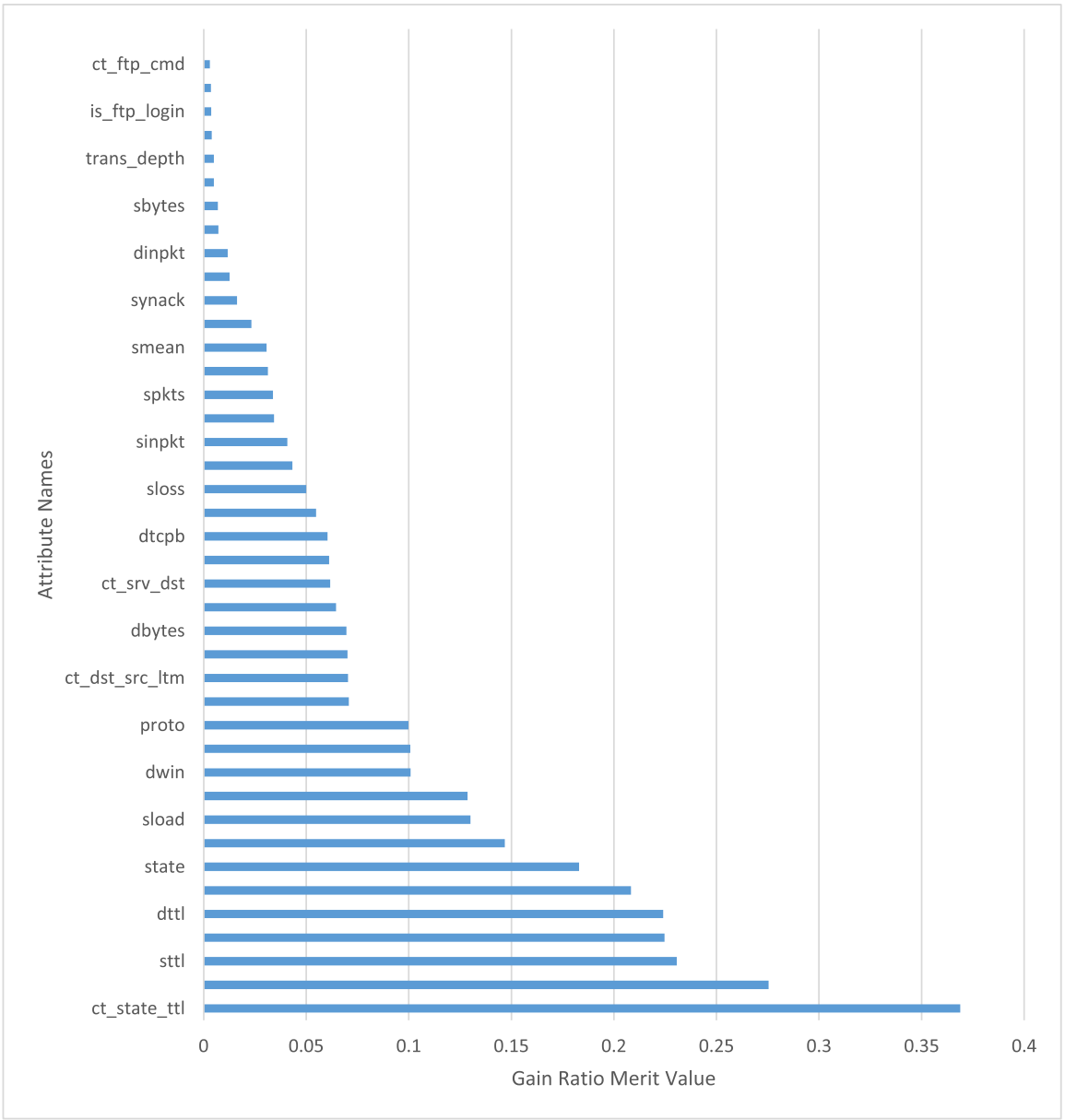


Fig. 4. Merit scores of features selected using gain ratio.

of 76.96.% and an error rate of 23.95% show that the IDS classification model is good and dependable. Also, the 79.80% precision of the developed classification model supports the reliability of the developed IDS classification. Consequently, MCC of 0.57 depicts the correlation coefficient between the exact and predicted values is fairly okay.

Conclusion

In this paper, ANN based IDS has been developed and evaluated on UNSW-NB15 intrusion detection dataset. Binarization discretization technique was used on continuous attributes and Gain ratio in ranking attributes. The first 30 attributes were selected based on adopted threshold prior to model building and evaluation using ANN-MLP. Experimental results revealed that the model gives an accuracy of 76.96% and MCC of 0.57 which shows a positive correlation. The results show that the technique adopted is promising and could be used for real time intrusion detection. The results also show that UNSW-NB15 is an efficient dataset to be used for network IDS.

However, this paper fails to test the performance of the model using different number of attributes. Future study will involve model performance with different attributes and adopts other preprocessing techniques and fast classifications techniques practicable for real time detection of intrusions on computer networks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] A.O. Adetunmbi, A PhD Dissertation submitted to the Department of Computer Science, Federal University of Technology, 2008.
[2] A.O. Adetunmbi, B.K. Alese, O.S. Ogundele, S.O. Falaki, A data mining approach to network intrusion detection, J. Comput. Sci. Appl. 14 (2) (2007) 24–37.
[3] AlSallal M. (2017), An integrated approach for intrinsic plagiarism detection. https://Securityintelligence.com retrieved on 24/9/2019
[4] R.V. Boppana, X. Su, On the effectiveness of monitoring for intrusion detection in mobile ad hoc networks, IEEE Trans. Mob. Comput. 10 (8) (2011) 1162–1174.
[5] M. Ennert, E. Chovancová, Z. Dudláková, Testing of IDS model using several intrusion detection tools, J. Appl. Math. Comput. Mech. 14 (1) (2015) 55–62.
[6] N. Garg, Binarization techniques used for grey scale images, Int. J. Comput. Appl. 71 (1) (2013) 8887–9975.
[7] F. Guo, Q. Zhao, X. Li, X. Kuang, J. Zhang, Y. Han, Y.A. Tan, Detecting adversarial examples via prediction difference for deep neural networks, Inf. Sci. 501 (2019) 182–192.
[8] KDDCup1999. Available-on:http://kdd.ics.uci.edu/databases/kddcup99/KDDCUP99.html, 2007
[9] Mahoney, Philip, An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection, Recent Advances in Intrusion Detection, Springer, Berlin Heidelberg, 2003.
[10] J.O. Mebawondu, O.J. Mebawondu, A.N. Atsanan, M.N. Suleiman, The impact of information technology on poverty alleviation in Nigeria, Continental J. Information Technology. 6 (2012) 1–15, doi:10.5707/cjit.2012.6.1.1.15.
[11] J. Mebawondu, Development of a Network Intrusion Detection System Using Neural Network M.Tech, Federal University of Technology, 2018.
[12] D. Mickler, G.M. Wachira, The AU's African governance architecture and SDG 16: examining intersections, in: Africa and the Sustainable Development Goals, Springer, Cham, 2020, pp. 49–57.
[13] N. Moustafa, J. Slay, The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 Data set, Inf. Secur. J. 25 (1-3) (2016) 18–31.
[14] N. Moustafa, J. Slay, UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set), in: Military Communications and Information Systems Conference (MilCIS), IEEE, 2015, pp. 1–6.
[15] B. Neethu, Classification of intrusion detection dataset using machine learning approaches, Int. J. Electron. Comput. Sci. Eng. 1 (3) (2015) 1044–1051.
[16] NSLKDD. Available on: http://nsl.cs.unb.ca/NSLKDD/, 2009.
[17] A.B Onyekwelu, B.K. Alese, A.O. Adetunmbi, Pre-processing of university webserver log files for intrusion detection, Int. J. Comput. Netw. Inf. Secur. (IJCNIS) 9 (1) (2017) 20–30.
[18] O.G. Opeyemi, O.S. Adewale, A.O. Adetunmbi, B.K. Alese, A.O. Ogunde, Deadlock detection in agent-based virtual knowledge communities, Ann. Comp. Sci. Ser. 8 (1) (2010) 27–44.
[19] O. Osanaiye, H. Cai, K.K.R. Choo, A. Dehghantanha, Z. Xu, M. Dlodlo, Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing, EURASIP J. Wirel. Commun. Netw. 1 (1) (2016) 130.
[20] J. Schmidhuber, Review deep learning in neural networks: an overview, journal homepage:, in: Neural Networks, 61, Elsevier Ltd, 2015, pp. 85–117. www.elsevier.com/locate/neunet.
[21] D.P. Vinchurkar, A Reshamwala, A review of intrusion detectiom system using neural network and machine learning technique, Int. J. Eng. Sci. Innov. Technol. 1 (2) (2012) 54–63 2012.
[22] B. Zhai, J. Chena, Development of a stacked ensemble model for forecasting and analyzing daily average PM2.5 concentrations in Beijing China, Science of The Total Environment 635 (2017) 644–658.