Ensemble and Deep-Learning Methods for Two-Class and Multi-Attack Anomaly Intrusion Detection: An Empirical Study

Adeyemo Victor Elijah[1], Azween Abdullah[2], NZ JhanJhi[3], Mahadevan Supramaniam[4], Balogun Abdullateef O[5]
School of Computing and IT, Taylor's University, Subang Jaya, Selangor, Malaysia[1, 2, 3]
Research and Innovation Management Centre, SEGI University, Malaysia[4]
Department of Computer Science, University of Ilorin, Ilorin, Kwara State, Nigeria[5]

***Abstract*—Cyber-security, as an emerging field of research, involves the development and management of techniques and technologies for protection of data, information and devices. Protection of network devices from attacks, threats and vulnerabilities both internally and externally had led to the development of ceaseless research into Network Intrusion Detection System (NIDS). Therefore, an empirical study was conducted on the effectiveness of deep learning and ensemble methods in NIDS, thereby contributing to knowledge by developing a NIDS through the implementation of machine and deep-learning algorithms in various forms on recent network datasets that contains more recent attacks types and attackers' behaviours (UNSW-NB15 dataset). This research involves the implementation of a deep-learning algorithm–Long Short-Term Memory (LSTM)–and two ensemble methods (a homogeneous method–using optimised bagged Random-Forest algorithm, and a heterogeneous method–an Averaged Probability method of Voting ensemble). The heterogeneous ensemble was based on four (4) standard classifiers with different computational characteristics (Naïve Bayes, kNN, RIPPER and Decision Tree). The respective model implementations were applied on the UNSW_NB15 datasets in two forms: as a two-classed attack dataset and as a multi-attack dataset. LSTM achieved a detection accuracy rate of 80% on the two-classed attack dataset and 72% detection accuracy rate on the multi-attack dataset. The homogeneous method had an accuracy rate of 98% and 87.4% on the two-class attack dataset and the multi-attack dataset, respectively. Moreover, the heterogeneous model had 97% and 85.23% detection accuracy rate on the two-class attack dataset and the multi-attack dataset, respectively.**

Keywords—Cyber-security; intrusion detection system; deep learning; ensemble methods; network attacks

I. Introduction

The proliferation of information and the technology used for enabling communication in everyday life has prompted the immense need for computer security [1]. The impact of Information and Communication Technology on economic growth, social wellbeing, private and public business growth, and national security is enormous as it provides the devices that propagate digital communications among hosts. The overall protection of these hosts, which exist as computers, network devices, network infrastructures, etc. [2], as well as data and information against cyber-attacks, worms, potential leakage and information theft is fundamental to cyber-security [3].

The level of research on the development of Intrusion Detection System (IDS) continues to increase as attacks abound and attackers continue to evolve in practice. As a result, IDSs must evolve to prevail over the dynamic malicious activities carried out over a network. The development of a Network Intrusion Detection System (NIDS) is critical for monitoring the network pattern behaviour of a computer networked system [4]. Typically, an IDS monitors network packets to facilitate the identification of attacks and are basically categorised as either misuse/signature or anomaly based. Signature based IDS matches attacks to previously known attacks, and anomaly-based IDS uses the created normal profile of a user to flag any profile that deviates from the user known behaviour [5].

Because of the unrelenting efforts of attackers to compromise a known network of computers and the new pattern of executing attacks and other malicious activities, the need for a robust, up-to-date IDS is imminent to adequately prevail against unknown attacks/threats or zero-day vulnerabilities.

As such, an empirical research study was conducted to develop an IDS that can address new types of attacks in our modern-day network using machine and deep learning algorithms. The contributions to knowledge produced during this research work are highlighted below:

1) The use of more recent and complex network data as input data, i.e. the UNSW-NB15 dataset, for the development of an IDS.

2) Two (2) methods of implementing ensemble learning methods for the development of an IDS;

3) Implementation of a deep-learning technique (LSTM) for building a NIDS;

4) Development of two (2) categories of NIDS, i.e., two-class (normal and attack labels) and multi-attack (ten class labels).

Moreover, it is the intent of this research work to answer the following research questions:

1) How effective is the ensemble learning method implementation of NIDS for detecting attacks, both in a two-class scenario and a multi-attack scenario?

2) How effective is the deep-learning implementation of NIDS for detecting attacks, both in a two-class scenario and a multi-attack scenario?

3) What peculiarities are found in two-class and multi-attack datasets and how do they affect the developed NIDS models?

II. Related Works

The research conducted by [6] presented a deep-learning method for developing a NIDS. The work proposed and implemented a Self-taught Learning (STL) deep-learning based technique on a NSL-KDD dataset. The STL model when evaluated based on training and test data achieved, in terms of percentage, 88.39% accuracy for 2-class and 79.10% accuracy for 5-class.

The work of [4] is a closely related work, wherein the authors developed a multi-classification NIDS using the UNSW-NB15 dataset and implemented an Online Average One Dependence Estimator and an online Naïve Bayes with 83.47% and 69.60% accuracy, respectively.

Another research work conducted by [7] reported the use of a deep neural network for development of a NIDS. The study implemented LSTM- Recurrent Neural Network (RNN) to identify network behaviour as normal or affected based on the past observations. KDDCup'99 was used as the dataset, and the work achieved a maximum value of 93% efficiency.

The research work carried out by [8] developed four different IDS models using the RNN algorithm and tested them on a NSL-KDD dataset (binary and 5-classes) to evaluate the models. The best model on a binary class achieved 98.1% accuracy using a 1-hidden layer BLSTM. For a 5-class, 87% accuracy was achieved using a 1-hidden layer BLSTM.

Using deep autoencoder (AE) after extracting features via statistical analysis methods, [9] developed an IDS that achieved 87% accuracy on NSL-KDD dataset.

The study of [10] focused on using machine learning methods for developing an IDS using J48, MLP and Bayes Network (BN) algorithms to achieve the overall best accuracy of 93% with J48, 91.9% accuracy using MLP and accuracy of 90.7% using BN on the KDD dataset.

III. Method

A. Dataset

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the US-letter paper size. If you are using A4-sized paper, please close this file and download the file "MSW_A4_format".

Most research studies on the development of IDS use the KDDCup'99 dataset; however, this dataset is gradually becoming (if not already) obsolete because it does not contain most new forms of attacks prevalent in modern networks of computers. Reflection of contemporary threats and the inclusion of normal network packets are two important features of a high-quality NIDS dataset. Because attackers execute dynamic attacks daily, it is thus necessary to make use of a recent dataset to uncover new malicious activities in a network [11]. Thus, UNSW-NB15 was used in this study. The UNSW-NB15 data was developed using the IXIA PerfectStorm tool in the Cyber Range laboratory of the Australian Centre for Cyber Security, which captured the sets of abnormal and modern-day normal network traffic. More details regarding the dataset creation are given in [2].

Table I provides insights into the datasets used in this study.

As depicted in Table I above, the dataset is comprised of 45 attributes, of which, two (2) are dependent variables. Two subsets of data are obtainable from the original dataset according to the dependent variables; one of these subsets was obtained to develop a two-class anomaly IDS, and the other was use dot develop a multi-attack anomaly IDS. The distribution of the attacks is contained in the attack_cat attribute, and the label attribute is comprised of normal and attack instances, denoted as 0 and 1, respectively.

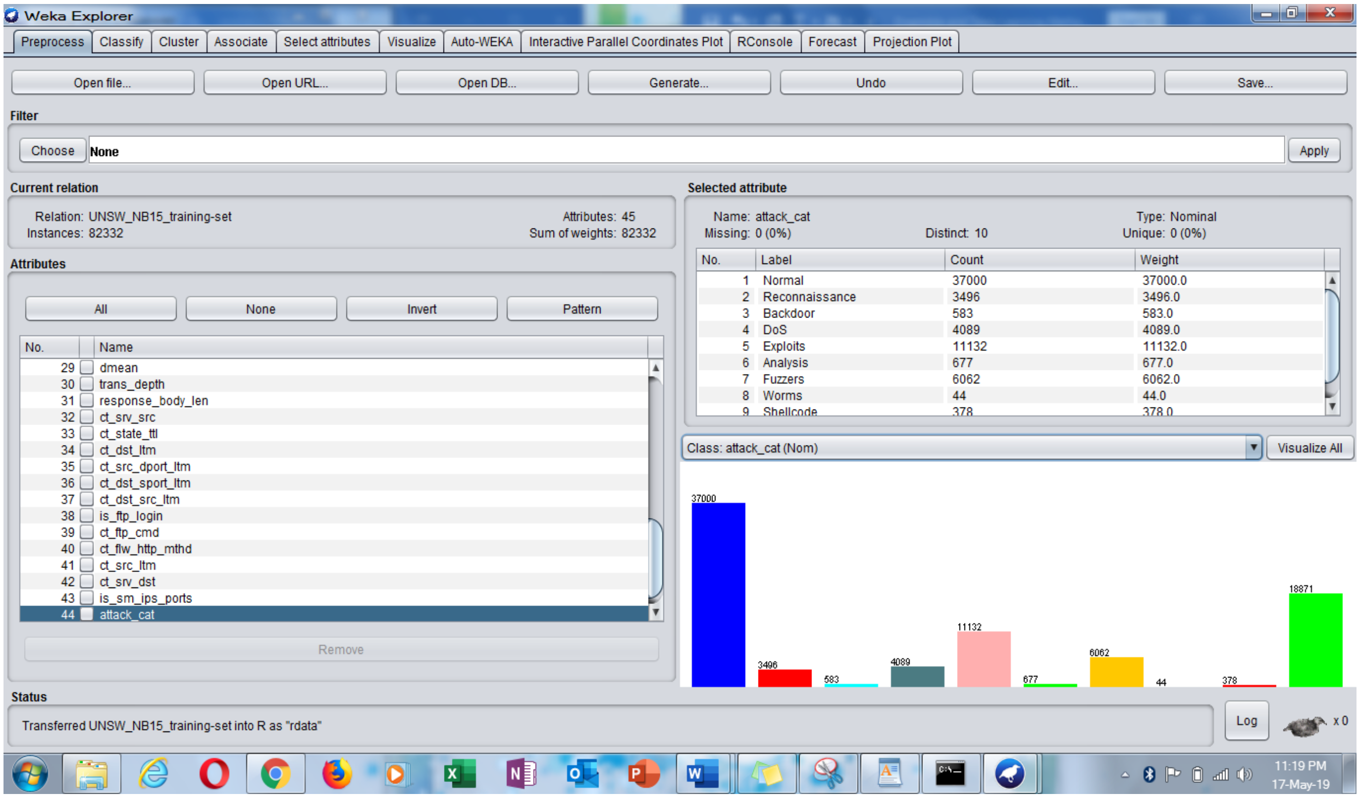
Regarding the features, Table II presents the details of both the independent and target variables.

Moreover, in light of data pre-processing and removal of redundant attributes, the first attribute indexed–id, serving as the index of the dataset, was removed because it is irrelevant, thus leaving two-class and multi-attack datasets with 43 attributes each.

Fig. 1 and Fig. 2 above depict the data distribution for both subsets of the original dataset. Fig. 1 depicts the ten (10) class labels of the multi-attack dataset presented in Table I; each of the labels is displayed using different colour. Fig. 2 shows the two-class labels as presented in Table I above, with blue colour representing the normal labels and red colour representing the attack labels.

TABLE. I. Dataset Description

Dataset Description		
No. of Attributes		45
No. of Independent Variables		43
No. Of Dependent Variables		2
Details of the First Dependent Variable	Name: label	
	Normal	Attack
	37,000	45,332
Details of the Second Dependent Variable	Name: attack_cat	
	Normal	37,000
	Reconnaissance	3, 496
	Backdoors	583
	DoS	4,089
	Exploits	11,132
	Analysis	677
	Fuzzers	6,062
	Worms	44
	Shellcode	378
	Generic	18,871



Filter

None

Current relation

Relation: UNSW_NB15_training-set   Attributes: 45
Instances: 82332   Sum of weights: 82332

Attributes

No.	Name
29	dmean
30	trans_depth
31	response_body_len
32	ct_srv_src
33	ct_state_ttl
34	ct_dst_ltm
35	ct_src_dport_ltm
36	ct_dst_sport_ltm
37	ct_dst_src_ltm
38	is_ftp_login
39	ct_ftp_cmd
40	ct_flw_http_mthd
41	ct_src_ltm
42	ct_srv_dst
43	is_sm_ips_ports
44	attack_cat

Selected attribute

Name: attack_cat   Type: Nominal
Missing: 0 (0%)   Distinct: 10   Unique: 0 (0%)

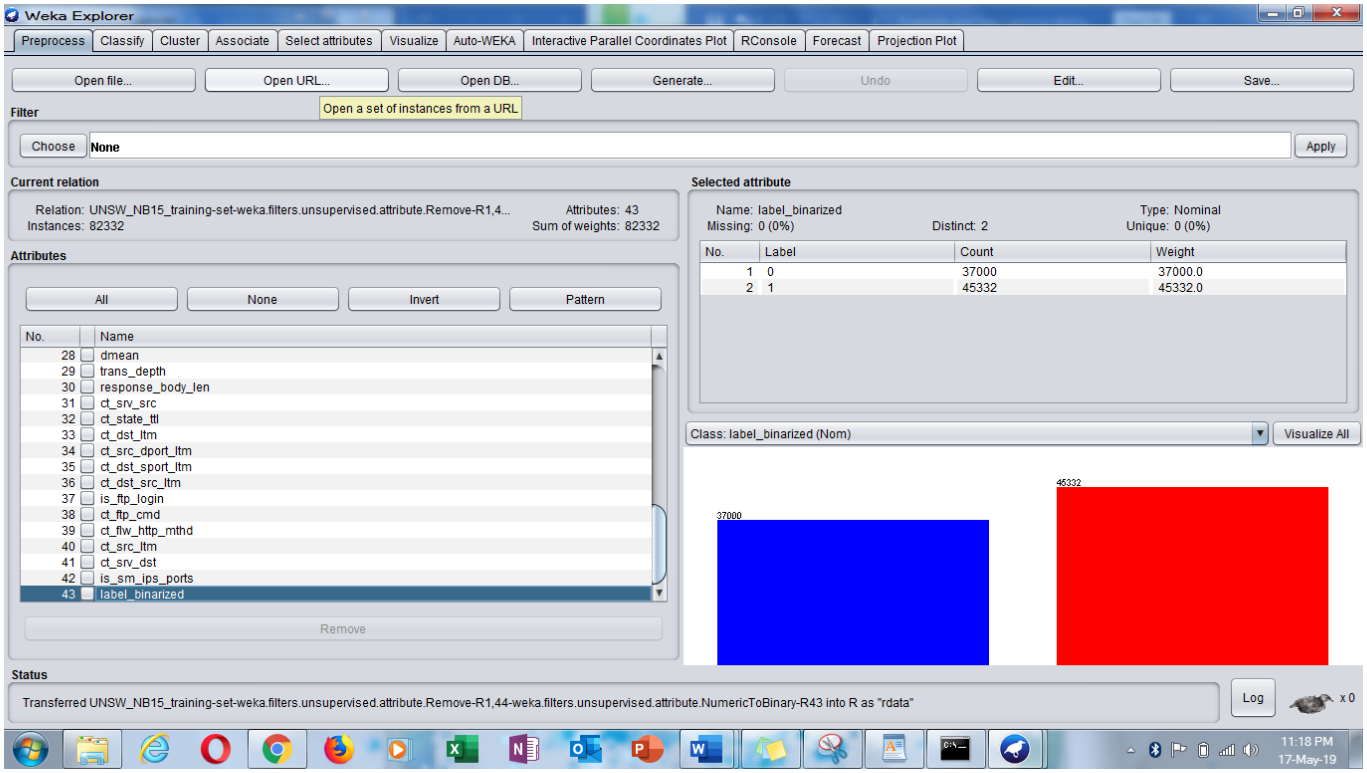
No.	Label	Count	Weight
1	Normal	37000	37000.0
2	Reconnaissance	3496	3496.0
3	Backdoor	583	583.0
4	DoS	4089	4089.0
5	Exploits	11132	11132.0
6	Analysis	677	677.0
7	Fuzzers	6062	6062.0
8	Worms	44	44.0
9	Shellcode	378	378.0

Class: attack_cat (Nom)

Status

Transferred UNSW_NB15_training-set into R as "rdata"

Fig. 1. Data Distribution in the Multi-Attack Dataset.



Filter

None

Current relation

Relation: UNSW_NB15_training-set-weka.filters.unsupervised.attribute.Remove-R1,4...   Attributes: 43
Instances: 82332   Sum of weights: 82332

Attributes

No.	Name
28	dmean
29	trans_depth
30	response_body_len
31	ct_srv_src
32	ct_state_ttl
33	ct_dst_ltm
34	ct_src_dport_ltm
35	ct_dst_sport_ltm
36	ct_dst_src_ltm
37	is_ftp_login
38	ct_ftp_cmd
39	ct_flw_http_mthd
40	ct_src_ltm
41	ct_srv_dst
42	is_sm_ips_ports
43	label_binarized

Selected attribute

Name: label_binarized   Type: Nominal
Missing: 0 (0%)   Distinct: 2   Unique: 0 (0%)

No.	Label	Count	Weight
1	0	37000	37000.0
2	1	45332	45332.0

Class: label_binarized (Nom)

Status

Transferred UNSW_NB15_training-set-weka.filters.unsupervised.attribute.Remove-R1,44-weka.filters.unsupervised.attribute.NumericToBinary-R43 into R as "rdata"

Fig. 2. Data Distribution in the Two-Class Dataset.

TABLE. II. UNSW-NB15 ATTRIBUTES

No.	Features	No.	Features
1	id	23	dtrcpb
2	dur	24	dwin
3	Proto	25	tcprtt
4	Service	26	synack
5	State	27	ackdat
6	spkts	28	smean
7	dpkts	29	dmean
8	sbytes	30	trans_depth
9	dbytes	31	response_body_len
10	rate	32	ct_srv_src
11	sttl	33	ct_state_ttl
12	dttl	34	ct_dst_ltm
13	sload	35	ct_src_dport_ltm
14	dload	36	ct_dst_sport_ltm
15	sloss	37	ct_dst_src_ltm
16	dloss	38	is_ftp_login
17	sinpkt	39	ct_ftp_cmd
18	dinpkt	40	ct_flw_http_mthd
19	sjit	41	ct_src_ltm
20	djit	42	ct_srv_dst
21	swin	43	is_sm_ips_ports
22	stcpb	44	attack_cat
		45	label

B. *Implemented Models*

This empirical analysis implements three (3) different data mining methods to develop a robust NIDS using both datasets mentioned above. The approaches include: (i) Homogeneous Ensemble, (ii) Heterogeneous ensemble, and (iii) Deep Learning (DL) implementations.

An ensemble method [12] is the process of combining some different results, produced by contributing base learners, of predictive models via different combination methods to make a final prediction based on aggregated learning. This method is typically implemented via two phases: the first phase being the construction of various models, and the second phase involving the combination of the estimates obtained from the various models [13]. The ensemble method is said to be homogeneous when the contributing base learners are multiples of the same computational characteristics (family). Base learners in an ensemble model are standard classifiers. In this study, the homogeneous ensemble was implemented in the form of the Random-Forest (RF) algorithm. The Random-Forest algorithm is a bagging method that consists of a finite number of decision tree algorithms with the addition of a ‘perturbation’ of the classifier used for fitting the base learners. In particular, RF uses ‘subset splitting’. The RF ensemble of trees makes use of only a random subset of the variables while building its trees; thus, the ensemble method is homogeneous.

Alternatively, a heterogeneous ensemble is the combination of various results of base learners that have different learning methods or computational characteristics, that is, the contributing base learners belong to different categories of classification algorithms. The standard classifiers for the heterogeneous ensemble considered in this study are described as follows: Bayes Theory (Naïve Bayes algorithm), Instance Learning (k Nearest Neighbour), Rule-based (RIPPER) and Tree methods (C4.5 Decision Tree). The voting combination method [14] [15] was adopted in this study for building the heterogeneous ensemble method. The voting method is a non-complicated method of combining several predictions of varied or different models, and it can be implemented in a variety of approaches, including majority vote, minority vote and average of probabilities. The average of probabilities method of voting [16] was selected for combining the results of each standard classifier because the averaged results of the models are used to provide the final prediction.

DL is an advanced implementation of a neural network. A neural network is the simulation of the human brain, that is, a model of connected neurons. A neural network is usually constructed to possess input, processing and output layers of neurons [17]. The processing layer, often referred to as the hidden layer, may contain one or more layers–a basic implementation of neural network is the Multilayer Perceptron (MLP) [18]. DL is an advancement on the MLP [19], but with more sophisticated and densely connected neurons that are capable of representing and extracting data in a more advanced form from data and mapping it into the output [20, 21]. The neural network implementations that are used for DL include but not limited to Convolutional Neural Network, RNN and Long Short-Term Memory (LSTM). In this study, the deep-learning method implemented was LSTM–a type of RNN. A typical LSTM [7] consists of a cell, an input, an output, and a forget gate, with which it captures the order dependence and recollection of values over random time intervals.

Using the three (3) different data mining methods discussed above, several predictive models were developed using the afore-mentioned datasets. Because it is known that model development is the next stage after the dataset and algorithm selection process and method identification phases, the percentage split model development process was used in this research work. The percentage split is the method of dividing a given dataset into two: the first part is used for executing a training phase-wherein the algorithms builds or fits their respective models, and the second part of the dataset is then used for testing–the phase whereby the fitted models are tested by making predictions using the independent variables of the disjoint test set. Thus, a certain percentage value is given to split the dataset into the training split and the test split. Moreover, having two datasets (two-class and multi-attack datasets), each selected algorithm was fitted on each dataset type, and the resulting models were tested on each corresponding test sets, thereby producing some sets of models that are categorised as (i) two-class attack anomaly IDS, and (ii) multi-attack anomaly IDS, each having three (3) separate models with respect to the applied method discussed above.

To summarise how the data mining methods were implemented and all robust NIDS models were all developed in this study, the proposed empirical framework is depicted in Fig. 3, and the experimental results produced are presented in table and charts and extensively discussed as seen in sections below.

C. Performance Evalutaion Metrics

Following the model development process stage, the developed models are evaluated. As such, the performances of models were evaluated based on the category they belonged to. The two-class anomaly IDS models were evaluated using the following metrics [17]: Detection rate, Area Under Curve (AUC), True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The multi-attack anomaly IDS models were evaluated based on the following metrics [18]: Detection rate, Kappa value and Weighted (AUC, TP, FP and F-measure). The multi-attack models were evaluated using weighted values because of the multiple values of the class labels (ten in number), unlike the two-class anomaly IDS, which has just two classes (normal and attack)–a binary classification model.

The proposed empirical framework presented in Fig. 3 above consists of the Data Pre-Processing and Re-Labelling Module and the Method Module, which interacts with the Model Development Process Module in producing the two forms of IDS mentioned in this study. The Algorithm module consists of the selected algorithms for this study, and this module interacts with the Method module, which defined the data mining implementations. Last, the Metrics component evaluates the produced model based on its form, and the evaluation results are subsequently discussed.

Table III presents the parameter settings for each algorithm used in this study. All models were trained and tested using the percentage split strategy–80% was used for training and 20% was used for testing, and their performances were evaluated using various metrics as appropriate for the type of the developed IDS model.

Conclusively, all experiments were carried using Waikato Environment for Knowledge Analysis (WEKA) tool for data analysis, wherein results were all obtained and presented in relevant section of this paper.

TABLE. III. IMPLEMENTATION OF EACH ALGORITHMS

Algorithm	**Parameter Settings**
NB	useKernelEstimator = True; useSupervisedDiscretisation = False, batchsize = 100
kNN	windowSize = 0, batchsize = 100
RIPPER	usePruning = True, seed = 1, batchsize = 100; folds = 5, minNo = 2.0; optimisations = 2, checkErrorRate = True
C4.5 Decision Tree	batchsize = 100, binarySplits = False collapseTree = True; confidenceFactor = 0.25; minNumObj = 2; numFolds = 5; subtreeRaising = True, unpruned = False; seed = 1. .useLaplace = False; useMDLcorrection = True
RF	bagSizePercent = 100; batchSize = 100; breakTiesRandomly = False; maxDepth = 0; computeAtrributeImportance = False; numFeatures = 30; numIterations = 20; seed = 1
LSTM (two-class)	reluAlpha = 0.01, Updater = adam, OptimizationAlgorithm = SGD, learning rate = 0.1, dataset= standardise. While developing the two-class anomaly IDS, LSTM layer was configured as neurons = 128, activation function = ReLU, gate activation function= Sigmoid, dropout = 0.3; Output layer parameter was lossFunction = LossMCXENT, activation function = softmax
LSTM (Multi-attack)	activation function = softmax; gate activation function = ReLU

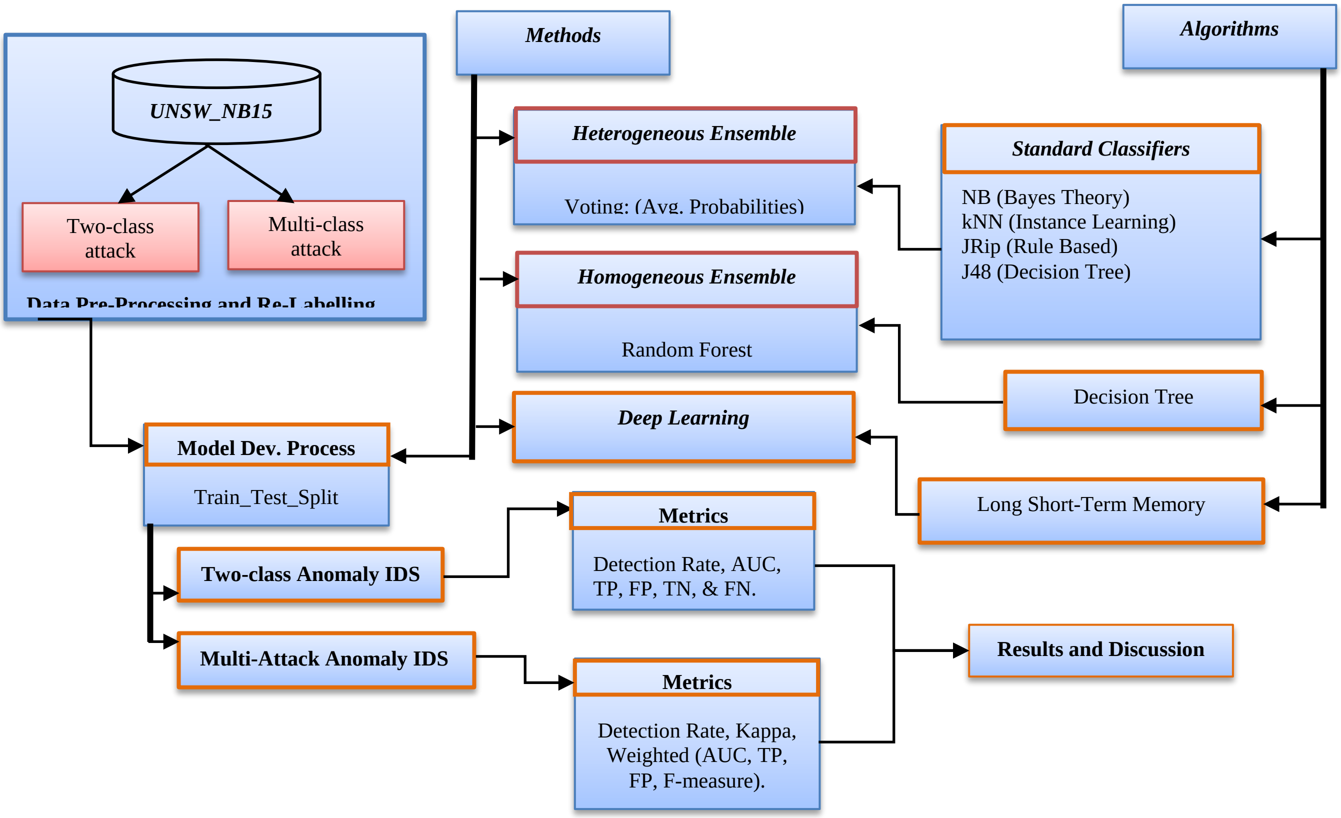


Fig. 3. Proposed Empirical Framework

IV. Results

Having implemented the proposed framework of this research, the reported results will be categorised into two according to the model development processes. Note that the test was conducted on 20% of the dataset, resulting in 16,466 instances. First, the two-class anomaly IDS is basically the prediction of whether a network packet is normal or an attack and is thus evaluated using the given metrics in Fig. 3. For the homogeneous method, Tables IV and V present the performance scores of the model and its corresponding confusion matrix, respectively.

From Table IV, the homogeneous ensemble had an overall detection rate of 97.96% with an AUC score of 0.997, indicating a very strong prediction model. The TP value of 0.98 indicates that the model classified 98% of normal packets as normal, and the TN value of 0.976 denotes that the attack packets were correctly classed as attack at the rate of 97.6%. The FP value of 0.024 denotes that just 2% of normal packets were classified as attack, and the FN value of 0.0158 indicates that approximately 1.58% of attack packets were predicted as normal. Likewise, Table V–the confusion matrix of the homogeneous ensemble, depicts the actual figures of the TP–7278 of 7395 normal instances classified as normal, FP–219 of 9071 attack instances misclassified as normal, TN–8852 of 9071 attack instances correctly classified as attack, and FN–117 of normal instances misclassified as attack.

For the heterogeneous ensemble, the voting cum average probabilities results for different techniques are shown in Tables VI and VII below.

TABLE. IV. Homogeneous Ensemble Model's Evaluation

Evaluation Metric	Score
AUC	0.997
TP Rate	0.984178
FP Rate	0.024143
TN Rate	0.975857
FN Rate	0.015822
Detection rate	0.979594

TABLE. V. Homogeneous Model Confusion Matrix

	Normal	*Attack*
Normal	7278	117
Attack	219	8852

TABLE. VI. Heterogeneous Ensemble Model's Evaluation

Evaluation Metric	Score
AUC	0.994
TP Rate	0.984043272
FP Rate	0.042994157
TN Rate	0.957005843
FN Rate	0.015956728
Detection rate	0.969148549

From Table VI, the heterogeneous ensemble had an overall detection rate of 96.92% with an AUC score of 0.994, indicating yet another very strong prediction model. The TP value of 0.98 indicates that the model classified 98% of normal packets as normal, and the TN value of 0.957 denotes that the attack packets were correctly classed as attack at the rate of 95.7%. The FP value of 0.43 denotes that approximately 5% of normal packets were classified as attack, and the FN value of 0.016 indicates that approximately 1.6% of attack packets were predicted as normal. Likewise, Table VII–the confusion matrix of the heterogeneous ensemble, depicts the actual figures of the TP–7277 of 7395 normal instances classified as normal, FP–390 of 9071 attack instances were misclassified as normal instances, TN–8681 of 9071 attack instances correctly classified as attack and FN–118 of 7395 normal instances misclassified as attack.

Last in this category, the results of deep-learning method for developing a two-class anomaly IDS as implemented with the specified parameters described in the previous section are shown in Tables VIII and IX.

Table VIII shows that the deep leaning model had an overall detection rate of 80.72% with an AUC score of 0.926, i.e. the deep-learning model is a competitive predictive model. The TP value of 0.57 indicates that the model classified 57% of normal packets as normal–a fair result as compared to other models in this category; it has a strong TN value of 0.998, indicating that the attack packets were correctly classed as attack at the rate of 99.8%-the best TN value in this category. The model had a FP value of 0.002, denoting an insignificant number of misclassified normal instances, and the FN value of 0.426 indicates that approximately 42.6% of attack packets were predicted as normal. Likewise, Table IX – the confusion matrix of the heterogeneous ensemble, depicts the actual figures of the TP–4239 of 7395 normal instances classified as normal, FP–19 of 9071 attack instances misclassified as normal, TN–9052 of 9071 attack instances correctly classified as attack and FN–3156 of normal instances misclassified as attack.

TABLE. VII. Heterogeneous Model Confusion Matrix

	Normal	Attack
Normal	7277	118
Attack	390	8681

TABLE. VIII. Deep-Learning Model's evaluation

Evaluation Metric	Score
AUC	0.926
TP Rate	0.573225
FP Rate	0.002095
TN Rate	0.997905
FN Rate	0.426775
Detection rate	0.807178

TABLE. IX. Deep-Learning Confusion Matrix

	Normal	Attack
Normal	4239	3156
Attack	19	9052

Critical evaluation of the models in this category reveals that, despite all models performing well using the AUC metric, the deep-learning model is weak in the detection of normal packets and will generate more false flagging of normal packets, thereby degrading the network monitoring in real time. Moreover, although the homogeneous and heterogeneous models competed fairly with each other, as they are both robust models for the detection of normal and attack packets, the homogeneous ensemble model is the best model in terms of lower FP and higher AUC values.

The second category is the multi-attack anomaly IDS, which is the classification of packets into normal and nine different types of attacks–a typical multi-classification problem, as discussed in previous section. The models are evaluated as depicted in Fig. 3. For the homogeneous ensemble method in this category, Table X reveals various performances scores.

Table X reveals the model's ability to detect whether a packet belongs to any of the ten (10) classes at 87.39%. This model had a kappa value of 0.8 and a weighted AUC of 0.98. The weighted TP value is 87.4%, and the weighted FP value is 2.5%. The model also had a weighted F-measure value of 0.87.

Similarly, in Table XI, this model detection rate was 85.23% but with a weighted AUC of 0.98, a weighted TP value of 0.852–85.2% correct classification of each class label instances, a weighted FP value of 0.031, a weighted F-measure of 0.855, and a kappa value of 0.79.

Last in this category, the deep-learning model of multi-attack anomaly IDS was also evaluated; its scores are represented in Table XII.

The deep-learning model yielded an ability to detect and predict the class of any packet at 72%. This result is achieved at a weighted AUC value of 0.868, a weighted F-measure score of 0.659, and a kappa value of 0.57. This model is capable of correctly detecting each class instance at the weighted TP value of 72.3, and it had a weighted FP value of 0.17.

TABLE. X. HOMOGENEOUS MODEL'S EVALUATION

Evaluation Metric	Score
Weighted AUC	0.98
Weighted TP Rate	0.874
Weighted FP Rate	0.025
Weighted F-Measure	0.87
Kappa Statistics	0.8227
Detection rate	87.3861

TABLE. XI. HETEROGENEOUS MODEL'S EVALUATION

Evaluation Metric	Score
Weighted AUC	0.982
Weighted TP Rate	0.852
Weighted FP Rate	0.031
Weighted F-Measure	0.855
Kappa Statistics	0.7928
Detection rate	85.2302

TABLE. XII. DEEP-LEARNING MODEL'S EVALUATION OF MULTI-ATTACK ANOMALY IDS

Evaluation Metric	Score
Weighted AUC	0.868
Weighted TP Rate	0.723
Weighted FP Rate	0.171
Weighted F-Measure	0.659
Kappa Statistics	0.57
Detection rate	72.26

In this multi-attack category, the homogeneous ensemble method also achieved the best performance, with a weighted F-measure of 0.87, a kappa value of 0.82, and an overall detection rate of 87%. Although the heterogeneous had a weighted AUC of 0.982, it is the second best in this category. Last, the deep-learning model competed fairly well with the other models, with its weighted AUC of 0.868; however, it had a low kappa value of 0.57 and a low detection rate of 72.26. Moreover, the confusion matrix for each model reveals the classification and misclassification of the instances accordingly. The deep-learning model was found to be unable to detect many attack classes, whereas the homogeneous model was adequately robust.

A summary of the detection rate of all models for both categories is presented in Table XIII.

Table XIII concisely presents the detection rates for all the above-described models, as is pictorially depicted in Fig. 4.

TABLE. XIII. SUMMARY OF THE RESULTS

Models	Methods	Detection Rate (%)
Two-class Anomaly IDS	Homogeneous Ensemble	97.96
	Heterogeneous Ensemble	96.92
	Deep Learning	80.72
Multi-attack Anomaly IDS	Homogeneous Ensemble	87.39
	Heterogeneous Ensemble	85.23
	Deep Learning	72.26

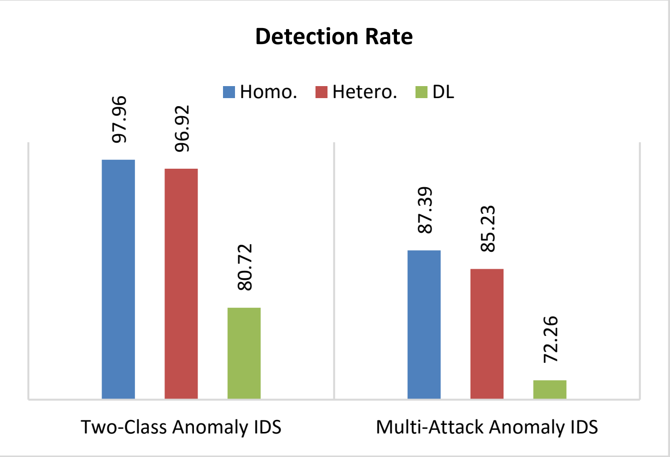


Fig. 4. Pictorial Representation of the Detection Rates of the Models.

V. Discussion

Based on the implementation of the various methods of machine learning and deep-learning algorithms in the development of several IDS models and making use of a modern-day dataset, it is possible to generalise the results. First, this study supports the fact that machine learning and DL are competent and effective technique for developing IDS in various capacities, such as two-class and multi-attack anomaly detection. This work also revealed that simple implementation of a machine learning algorithm is required and is a much more effective with less computational cost and complexity in the development of IDS regarding the strong predictive prowess of the homogeneous ensemble method compared to the heterogeneous (though it fiercely competed) and the deep-learning methods. Moreover, it can be generally stated that the detection rate for a two-class IDS is higher than that of multi-attack IDS because of the number of classes the machine learning will learn to make correct predictions and also the nature of data, wherein imbalance is peculiar to the multi-attack dataset, whereas the two-class dataset is mostly balanced.

Generally, the best model produced by this research work for detecting either a normal or attack packet (two-class anomaly IDS) operates at the rate of 97.96% and the best model for the multi-attack (ten-classes) anomaly IDS has the detection rate of 87.39%. In direct comparison with the recent work of [4], which actually outperformed much past research models, their work produced an overall detection rate of 83.47% for their online AODE model and a 69.60% detection rate for their online Naïve Bayes model, both of which were outperformed by the best (87.39%) and second-best (85.23%) detection rate of the multi-attack anomaly IDS models developed in this research work.

Comparatively, the research work conducted by [6] produced NIDS of 88.39% accuracy for a two class attack which was outperformed by two of the three NIDS of this study developed for 2-class attack detection (with 97.96% and 96.92% detection rate produced in this study), and also while their work produced a 79.10% accuracy for 5-class, the NIDS developed in this study produced two out of the three NIDS with 87.39% and 85.23% detection rate for 10-classes. Also, their STL model yielded a 75.76% f-measure value for the 5-class NIDS while this study produced 87% for homogeneous ensemble and 85% for heterogenous ensemble for a 10-class NIDS.

Additionally, the study of [10] developed IDS using J48, MLP and Bayes Network (BN) algorithms to achieve the overall best accuracy of 93% with J48, 91.9% accuracy using MLP and accuracy of 90.7% using BN on the KDD dataset. The homogenous ensemble NIDS developed in this study outperformed their work with a detection rate of 97.96% as well as the heterogenous ensemble NIDS with the detection rate of 96.92%

Having implemented several machine learning and deep-learning algorithms and several techniques for combining models, the application of feature selection technique to best select features from the available ones in the dataset is recommended as future work and also in practice to produce an optimal model with less cost and computational complexity. Moreover, the deep-learning method requires further investigation because there is need for improvement in both two-class and multi-attack anomaly IDSs.

VI. Conclusion

This research work revealed answers to several research questions. In response to the first question, the NIDS developed using machine learning is highly effective with a homogeneous ensemble implementation achieving a detection rate as high as 98% in a two-class scenario and 87% in a multi-attack scenario, and its heterogeneous counterpart is effective for NIDS with a detection rate of 97% in a two-class scenario and 85% in a multi-attack scenario.

In response to the second research question, the empirical research work revealed that a deep-learning implementation can be effective at as low as 80% detection rate in a two-class scenario and can effectively detect various types of attacks and normal packets in a multi-attack scenario at 72%.

Answering the third research question, it was discovered that two-class datasets have a balanced distribution unlike the multi-attack which is greatly imbalanced. These peculiarities affected the developed models as the developed models better fitted the balanced dataset than the imbalanced dataset.

The results of this research work also revealed that it is easier to identify two classes of network packet than ten (10) different classes belonging to a network packet.

This research work also revealed the weakness of DL, as it cannot produce a competitive model if its configuration is not sophisticated, i.e., is comprised of a high number of layers, which in turn increases computational complexity and cost.

A dataset consisting of 43 attributes is usually considered as a high-dimensional dataset that requires a feature pre-processing stage, wherein redundant, irrelevant and (in some cases) highly correlated attributes are removed to develop a robust model that neither over fits or under fits the dataset. This stage is executed by applying a feature selection technique, which includes filter and wrapper methods; however, this stage was not conducted in this research work and will be considered in future work. Additionally, while developing NIDS using two-class dataset, it was discovered that the dataset was imbalance. Thus, class balancing is also considered as future work.

The development and deployment of the developed NIDS models for real time detection of attack is considered as an important future work.

References

[1] M. Ahmed, A. N. Mahmood, and J. Hu, “A survey of network anomaly detection techniques,” J. Netw. Comput. Appl., pp. 1–13, 2015.

[2] N. Moustafa and J. Slay, “UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems,” Proc. ofthe Mil. Commun. Inf. Syst. Conf., pp. 1–6, 2015.

[3] A. O. Balogun, A. M. Balogun, V. E. Adeyemo, and P. O. Sadiku, “A Network Intrusion Detection System : Enhanced Classification via Clustering,” Comput. Inf. Syst. Dev. Informatics Allied Res. J., vol. 6, no. 4, pp. 53–58, 2015.

[4] M. Nawir, A. Amir, N. Yaakob, and O. N. G. B. I. Lynn, "Multi-Classification of Unsw-Nb15 Dataset for," vol. 96, no. 15, pp. 5094–5104, 2018.

[5] S. M. Thaler, Automation for information security using machine learning. 2019.

[6] Q. Niyaz, W. Sun, A. Y. Javaid, and M. Alam, "A Deep Learning Approach for Network Intrusion Detection System," 2016.

[7] M. Ponkarthika and V. R. Saraswathy, "Network Intrusion Detection Using Deep Neural Networks," vol. 2, no. 2, pp. 665–673, 2018.

[8] A. Elsherif, "Automatic Intrusion Detection System Using Deep Recurrent Neural Network Paradigm," 2018.

[9] C. Ieracitano, A. Adeel, M. Gogate, K. Dashtipour, and C. R. Aug, "Statistical Analysis Driven Optimized Deep," Int. Conf. Brain Inspired Cogn. Syst. Springer, Cham., pp. 759–769, 2018.

[10] M. Alkasassbeh and M. Almseidin, "Machine Learning Methods for Network Intrusion Detection," no. October, 2018.

[11] G. Li and Z. Yan, "Data Fusion for Network Intrusion Detection : A Review," vol. 2018, 2018.

[12] G. Seni and J. F. Elder, Ensemble Methods in Data Mining: Improving Accuracy Through Combining Predictions. 2010.

[13] P. Illy, G. Kaddoum, C. M. Moreira, K. Kaur, and S. Garg, "Securing Fog-to-Things Environment Using Intrusion Detection System Based On Ensemble Learning," no. April, pp. 15–18, 2019.

[14] M. Sabzevari and G. Mart, "Vote-boosting ensembles," Pattern Recognit., 2018.

[15] D. Murphree et al., "Ensemble Learning Approaches to Predicting Complications of Blood Transfusion," in IEEE Eng Med Biol Soc., 2016, pp. 1–11.

[16] I. H. Witten, E. Frank, and M. A. Hall, Data Mining - Practical Machine Learning Tools and Techniques. 2011.

[17] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," IEEE Access, vol. 6, pp. 35365–35381, 2018.

[18] M. A. Mabayoje, A. O. Balogun, A. O. Ameen, and V. E. Adeyemo, "Influence of Feature Selection on Multi - Layer Perceptron Classifier for Intrusion Detection System," Comput. Inf. Syst. Dev. Informatics Allied Res. J., vol. 7, no. 4, pp. 87–94, 2016.

[19] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: A plug and play device," Ad Hoc Networks, 2018.

[20] D. Papamartzivanos and G. Kambourakis, "Introducing Deep Learning Self-Adaptive Misuse Network Intrusion Detection Systems," IEEE Access, vol. 7, 2019.

[21] SH Kok, Azween Abdullah, NZ Jhanjhi, Mahadevan Supramaniam, "A Review of Intrusion Detection System Using Machine Learning Approach", in International Journal of Engineering and Research, Jan 2019.