Evaluation of Selected Stacked Ensemble Models for the Optimal Multi-class Cyber-Attacks Detection

Olasehinde Olayemi Oladimeji[1], Alese Boniface Kayode[2], Adetunmbi Adebayo Olusola[2] & Aladesote Olomi Isaiah[3]

[1]Department of Computer Science, Federal Polytechnic, Ondo State, Nigeria.
[2]Department of Cyber Security, Federal University of Technology, Akure, Ondo State, Nigeria.
[3]Faculty of Computer Science and Information Technology, University Putra Malaysia.

ABSTRACT

The significant rise in the frequency and sophistication of cyber-attacks and their diversity necessitated various researchers to develop strong and effective approaches to address recurring cyber threat challenges. This study evaluated the performance of three selected meta-learning models for optimal multi-class detection of cyber-attacks using the University of New South Wales 2015 Network benchmark (UNSW-NB15) Intrusion Dataset. The results of this study show and confirm the ability of the three base models; Naive Bayes, C4.5 Decision Tree, and K-Nearest Neighbor for solving multi-class problems. It further affirms the knack of the duo of feature selection techniques and stacked ensemble learning to optimize ML models' performances. The stacking of the predictions of the information gain base models with Model Decision Tree meta-algorithm recorded the most improved and optimal cyber-attacks detection accuracy and Mattew's correlation Coefficient than the stacking with the Multiple Model Trees (MMT) and Multi Response Linear regression (MLR) Meta algorithms.

Keywords: Cyber-attacks, Base model, stacked ensemble, Meta Learners, Evaluation, performance improvement, Intrusion, Feature selection

1 INTRODUCTION

The heavy dependence on the internet throws up an enormous contest for individuals' security and organization information. There has been a significant rise in the frequency of cyber-attacks and their diversity. The industries and cyber communities are being faced with new kinds of attacks daily. Cyber-attacks' high complexity poses a threat to the protection network devices and the CIA triad of sensitive information stored on them. Many organizations admit that the daily compromise of sensitive information and computer security is complicated, and it is hard to have a cyber-system that is entirely free from attacks. These attacks' sophistication and relentlessness have induced various researchers to develop strong and effective approaches to address recurring cyber threat challenges. The use of heterogeneous hybrid models for cyber-attacks detection has proven to be more effective than single models [1]. Hybrid solutions sought to improve the model's detection accuracy to obtain optimal detection of cyber-attacks detection. This paper is an extension of [1]. While the work in [1] focused on improving binary cyber-attacks detection, this work focuses on multi-class cyber-attacks detection improvement.

Cyber-attacks are malicious events directed against cyber devices to compromise them and their contents. Intrusion always precedes cyber-attacks; Olasehinde et al. [2] defines an intrusion as any or set of processes or actions that bypass or fools authentication or subvert access control procedures. It also categorized cyber-attacks into three categories: Scanning attacks, Seizure attacks, and Penetration attacks. Scanning attacks are information gathering network attacks in quest of the status and vulnerabilities of the hosts and the network [3]. It sent a port scan (probe) to ascertain the host system's strength/weakness and scrutinized its response to uncover the target system's characteristics and vulnerability. Scanning attacks are generally used to detect a potential victim; scanning is more than a type of attack. It is also the first phase of seizure and penetration attacks. Seizure attacks are Claim-and-hold attacks; it legitimately grasps system resources and declines to release them for other users who need them, resulting in a seizure of the computer resources and denying service to legitimate users. Denial of Service (DoS) and Ransomware attacks are good examples of seizure attacks. Penetration attacks exploit imperfection in software design and development to modify and alter the state of the system. It installed malware and viruses on the compromised system to gain unauthorized control of the system. Hackers often employ penetration (attacks) testing to gain unauthorized access to the host without the administrator's consent.

Machine Learning (ML) is an artificial intelligence (AI) approach that provides systems the ability to learn patterns of attacks and normal packets from captured network traffic [4] or intrusion dataset and use the acquired intelligence to build intrusion detection system (IDS). ML's goal is to make an excellent guess useful to the predictive (classification) problem. Supervised ML algorithms extract valuable knowledge from the mapping of supplied inputs and its desired output (class label) of the training dataset, then validates the testing dataset's obtained knowledge. Regression and classification are examples of supervised's ML techniques. The classification problem is termed binary when it involves two possible output (labels) and multi-class when the output labels are more than two. Unsupervised learning draws knowledge from a dataset consisting of input data without label responses. It partitions the dataset into clusters based on the similarities that exist among the dataset. It validates by assigning a new test instance into the appropriate cluster; clustering analysis and association mining are examples of unsupervised learning methods.

IDSs are network security tools and predictive models used to monitor, analyze, and classify network traffics as either normal or attack. IDSs are used to protect the cyber system and network against hackers and cyber attackers' nefarious activities. IDSs are categorized according to where the intrusion is detected (within a host or in the network) and the type of detection method used (Anomaly, Signature, and Hybrid). The anomaly detection method, also known as behavioral-based detection, measures and stores normal traffics behavior patterns. Any shift or deviation from the established normal practice of action is classified as an intrusion (attacks). The anomaly method can detect unknown attacks but generates a higher false alarm rate. The signature-based detection method, also known as misuse detection methods, uses the prior knowledge of known attack signatures to detect intrusion. It cannot detect attacks with an unknown signature. It is very good at detecting known attacks and generates a low false alarm rate. Hybrid detection methods combine misuse and anomaly detection methods to complement each other; any deviation in the pattern of network traffic will be detected by the anomaly engine, while the signature engine will detect all known intrusion (attack)

Several authors have proposed network attack detection solutions. In [5], five ML classifiers' capabilities on the NetFlow dataset were explored as a feasible method of detecting malicious traffic in a network. Random forest classifier recorded more than 95% detection accuracies in 8 out of the 13 scenarios set up in the experiments. Kumar et al. [6] implemented a mean-shift unsupervised clustering algorithm for detecting attacks in computer networks. The experimental results show the effectiveness of this method in detecting cyber-attacks with a detection accuracy of 81.2%. A Convolutional Neural Network (CNN) based multi-class attacks detection system was proposed in [7], a Genetic Algorithm (GA) was used to optimize the CNN classifiers to find a better layout of the input features. The CNN classifier's implementation on the UNSW-NB15 dataset with ten multi-class shows the accuracy of 98.14% and a Kappa coefficient of 0.6386 on the best six attack categories detected. The CNN classifier's performances on the NSL-KDD testing dataset recorded an accuracy of 94.47% for normal traffic detection and a Kappa coefficient of 0.67. Elmasry et al. [8] proposed a multi-class detection system using four deep learning models (deep neural networks, long short-term memory recurrent neural networks, gated recurrent unit recurrent neural networks, and deep belief networks) on four publicly available intrusion detection datasets: KDD CUP 99, NSL-KDD, CIDDS, and CICIDS2017. The results of this study show a significant network attack detection improvement. Authors in [9] compared the anomaly multi-class attack detection performances of four Apache Spark models of Decision Tree, Naive Bayes, Random Forests, and Multilayer Perceptron on the Supervisory Control and Data Acquisition (SCADA) dataset. Decision Tree and Random forest classifiers recoded a better detection performance than the rest of the classifiers.

IDS models (classifiers) ' performance depends on the classification algorithm and the network traffics attributes used to train the ML algorithms to build the IDS mode [2]. Feature Selection (FS) is a crucial dataset pre-processing techniques used to improve ML models' performance [10]. FS's goal is to select the relevant attributes of the network traffics or intrusion detection dataset positively correlated to the output attribute (class label) to train the ML algorithm to build an optimal IDS model [11]. The irrelevant and redundant features of the network traffics cannot contribute to the efficient and effective IDS models; it will only confuse the classifiers and lead to incorrect classification results. FS has demonstrated its ability to effectively enhance learning efficiency, improve predictive accuracy, reduce the complexity of learned results, and reduce model overfitting [12], [13], [14]. Filtered Based, Wrapper, and Embedded are three FS techniques. Filtered based FS techniques select relevant attributes based on statistical measures and independent of the learning algorithm. Filtered based FS is fast and computationally simple and can easily be scaled to high dimensional dataset [15]. Wrapper FS techniques select the best subset of attributes based on the result of the learning algorithm. The chosen attribute subset's quality is directly measured by the ML algorithm's performance applied to that attribute subset. It considers attribute dependencies [16]. Wrapper techniques are computationally intensive, slower but more accurate than the filter-based, and have high risks of models overfitting. They are selected based on statistical measures.

Embedded FS techniques employ hybrid and ensemble learning methods for the attributes selection. The decision on the attribute subset to be chosen is a collective one. Embedded FS techniques are built into the classifier construction; it is computationally expensive as wrapper methods. It performs better than the other two techniques. In [17], Support Vector Machine classifiers of three filter-based feature selection techniques, Chi-square, information, and Relief, were proposed for the multi-class classification of clothing review dataset taken from Kaggle. In [18], the authors applied filter-based attributes evaluation of the NSL-KDD dataset and the ensemble classifier of KNN, Random Tree, Rep Tree J48 Decision Tree, and Random Forest Base models. The ensemble model detects intrusion, minimized computational expense, and improved accuracy. Wrapper feature selection technique based on the firefly optimization algorithm and SVM was used to improve the IDS classifier's classification accuracy in the work in [19]. The work in [20] implemented a CFS-BA feature selection and voting ensemble-based IDS with NSL-KDD, AWID, and CIC-IDS2017 intrusion datasets. The ensemble model recorded improved detection accuracy.

Stacked ensemble employs Meta-learning algorithms to learn from and combines (build synergy among) the predictions of two or more IDS models to improve their combined IDS detection accuracy rate [1]. Meta-learners are supervised learning methods that learned systematically from previous experience [21]. It is good at learning from a different experience than the original learner [1]. Meta-learners combines the predictions of two or more base models generated by ML algorithms L_1,... L_n. on a single dataset D, which is made up of feature vector d_i = (x_i, y_i). The stacked ensemble framework first generates a set of base models h_i,......,h_n, where $h_i = L_i(D)$, then combines the base model predictions using the Meta level algorithm.

Several Authors have applied ensemble learning to improve the classification accuracies performance of IDS models. In [22], the authors used correlation feature selections with bagging and boosting ensemble models to improve the binary and multi-class packet detection rate and low false alarm rate for KDD and NSL-KDD datasets. Findings from work in [23] reveal that the stacked ensemble of ANN, SVM, RF, and CART is suitable for improving the detection accuracies of IDS models evaluated on the NSL-KDD dataset. In [24], the authors proposed using the stacking of RF, LR, KNN, and SVM-based models to improve network intrusion detection on UNSW NB15 and UGR 16 real-time packets; the experimental results of this work show a superior detection of a real-time dataset. ML classifiers' limitations in identifying malware due to the polymorphic, metamorphic, and zero-day malware behavior motivated the work in [25]. A stacked ensemble classifier of the C5.0 decision tree model and anomaly One-class Support Vector

Machine classifier was proposed to detect known intrusion and zero-day attacks. The stacked classifier's evaluation on the NSL-KDD and ADFA dataset shows an improved detection rate and low false alarm rate compared with the two base classifiers. Olasehinde et al. [26] implemented a Multiple Model Tree (MMT) Meta algorithm stacked ensemble classifier for the binary and Multi-class detection accuracy improvement UNSW-NB15 dataset. The stacked ensemble improves the detection accuracy and false alarm rate of the KNN, NB, and C4.5 DT base classifiers. This work, unlike the works in [8] and [9], employs second-level learning (stacked ensemble) to improve the detection performance of the ML's models. It also compares the three Meta algorithms used for the second level learning and model building. The models in [8] were evaluated on KDD CUP 99, NSL-KDD, CIDDS, and CICIDS2017 intrusion detection dataset, while the UNSW-NB15 intrusion dataset was used to evaluate the proposed work.

The researchers in [1] focused on binary cyber-attacks (Attacks/Normal) detection accuracy improvement of three stacked ensemble models of three base models of KNN, NB, and C4.5 DT. The experimental result revealed that MDT recorded the best binary improvement, while MMT performs relatively better than MLR. This study aimed to improve the multi-class cyber-attacks detection accuracies and comparatively evaluate the performances of the selected Stacked Ensemble Algorithms in [1] based on their strength to increase the base learners' multi-class detection accuracy rate. This study has the following contributions:

i. We first highlight feature selection and ensemble learners' ability to improve ML's cyber-attack detection models' performances.

ii. We then present three stacked ensemble IDS models for multi-class attack detection improvement of cyber network attacks.

iii. Finally, we evaluated the Multi-class attacks detection accuracy and F1-score performance metrics for both the base and stacked ensemble models. We also computed Matthew's Correlation Coefficient (MCC) for each model. We determined the attacks detections improvement of the stacked ensemble models over the base models in terms of multi-class detection accuracy and models' MCC score. The experimental results show that all the stacked ensemble models significantly improve multi-class cyber-attack detection on the UNSW-NB15 intrusion dataset.

2. Methodology

The proposed architecture of the Optimal Multi-class Cyber Attacks Detection Evaluation of Selected Stacking Algorithms is depicted in Figure 1. It consists of two sections; the first section is the stacked ensemble model building section. It is indicated with the continuous black arrow lines in figure

1. It comprises three phases; the pre-processing phase involves the discretization and selection of the UNSW-NB15 dataset's relevant features. Three filtered-based feature selection techniques; (Information gain, correlation, and consistency) were used to select the UNSW-NB15 intrusion dataset's relevant features. The whole feature dataset and the reduced features datasets were used to train and evaluate the K Nearest Neighbor, C4.5 Decision Tree, and Naive Bayes' base algorithms via ten-fold cross-validation in the second phase. In the third phase, the evaluated predictions of the base models in the second phase were used to train and build the stacked ensemble models of the three selected Meta algorithms: Multiple Model Tree (MMT), Multi Response Linear Regression (MLR), and Meta Decision Tree (MDT).

The second section is the model's evaluation section. It is indicated with the short red dashed arrow lines in figure 1. The testing dataset was used to evaluate each of the three reduced features base models and their stacked ensemble models. We use the Python Programming language to implement the cyber-attack detection models.

2.1 UNSW-NB15 Dataset

The UNSW NB-15 dataset was developed using the IXIA Perfect Storm tool by the cybersecurity research group at the Australian Center for Cyber Security [27]. It is a fusion of normal network traffic packets, and synthetic modern-day network traffics attacks. The training and testing contain 82,332 and 174,341 records with 49 features each, respectively [27]. The dataset comprises nine attack categories and normal traffic, and it is suitable for the effective detection of existing and new attacks [28]. The details of both attack and normal traffic, coupled with the records in the training and testing categories, are presented in Table 1.

Table 1: Names and No of Attacks Categories in the UNSW-NB15 Dataset

	Training		Testing	
Names of Attack	No of Connection	Percentage Distribution	No of Connection	Percentage Distribution
Reconnaissance	3496	4.25	10491	5.98
Dos	4089	4.9	12264	6.99
Exploit	11132	13.52	33393	19.04
Shellcode	378	0.46	1133	0.65
Fuzzers	6062	7.36	18184	10.37
Backdoor	583	0.71	1746	1.00
Analysis	672	0.82	2000	1.14
Generic	18871	22.92	40000	22.81
Worms	44	0.05	130	0.07
Total No of Attacks	45332	55.06	119341	68.06

Normal	37000	44.94	56000	31.94
Total No of Connections	82332	100.00	175341	100.00

2.2 Data Munging and Analytic

This section outlines the discretization method, feature selection techniques, base, and meta-algorithms analytic techniques used for this study.

2.2.1 Discretization and Feature Selection Techniques

The discretization process converts the continuous numerical values of dataset attributes into discrete values suitable for machine learning. Discretization reduces demands on system memory and data storage spaces. It improves the efficiency of data mining and makes machine learning faster and accurate.[29] The Class Attribute Interdependent Maximization (CAIM) discretization method was used to discretize the UNSW-NB15 dataset to make it suitable for the data analytic. The Discretized variable D for attribute F of the dataset is given in equation

$$1(C, D \mid F) = \frac{\sum_{i=1}^{n} \frac{max_i^2}{m_{ir}}}{n} \qquad (1)$$

where n is the user's predefined number of intervals, i iteration through all intervals. i=1,2,...,n, max is the maximum value within the ith column; Mir is the total number of continuous values of attribute F.

Three filter-based (mutual information) feature selection techniques were used to select the most relevant features of the UNSW-NB15 dataset. The correlation technique uses the merit function, as shown in equation 2, to choose the dataset's most pertinent attribute. Given all the feature subsets of the UNSW-NB15 dataset, Merit (M_s) is evaluated for each subset, the feature subset with the highest merit score is selected.

$$M_s = \frac{k\bar{r}_{cf}}{\sqrt{k + k(k-1)\bar{r}_{ff}}} \qquad (2)$$

where $\bar{r}_{cf}$ is the average attack categories to features, $\bar{r}_{ff}$ is the average features to features correlations, and k is the number of features in the subset S

Given a training sample S, the inconsistency count (IC) of an instance of subset A ∈ S is given in equation 3

$$\mathrm{IC}_{X'}(A) = X'(A) - \max_{k} X'_k(A) \qquad (3)$$

$X'(A)$ is the number of instances in S equal to subset A using only the features in $X'$and $X'_k(A)$is the number of instances in S of class k equal to A using only the features in X'.

The consistency features techniques select features subset with the lowest inconsistency rate as shown in equation 4

$$IR(X') = \frac{\sum_{A \in S} IC_{X'}(A)}{|S|} \qquad (4)$$

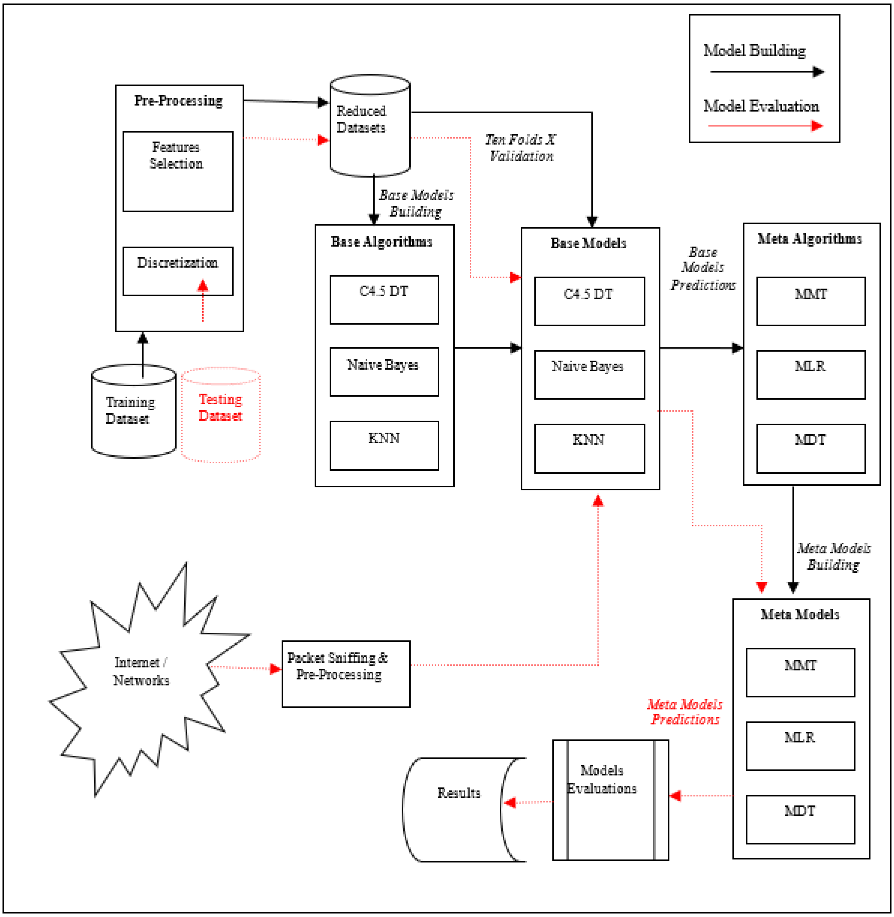

Figure 1: Evaluation of Selected Stacked Ensemble Models for the Optimal Multi-class Attacks Detection

The Information Gain attributes selection scores and rank attributes based on their information gain to the target feature (Y), Information Gain (IG) for attribute x is given in equation 5

$$IG(X) = H(Y) - H(Y|\,X) \qquad (5)$$

Where H(Y) is Entropy of Y as shown in equation 6 and H(Y|X) is Entropy of Y given X as shown in equation 7

$$H(Y) = -\sum_{i=1}^{n} p(y_i)\, log_2 p(y_i) \qquad (6)$$

$$H(Y \mid X) = -\sum_{i=1}^{n} p(x_i) \sum_{j=1}^{k} p(y_j \mid x_i) log_2 p(y_j \mid x_i) \quad (7)$$

Where n: is the number of instances in the dataset. k: is the number of attacks in the UNSW-NB15 dataset P(y_i): is the probability of occurrence of an attack-type of an instance i and, P($y_i|x_i$) is the probability of an attack-type of an instance i, given the occurrence of feature value x of instance i

2.2.2 Data Analytic Tools

The three Meta learner algorithms for evaluation in this study, namely; MLR, MMT, and MDT, were used to build stacked ensembles individually of the predictions of C4.5 DT, NB, and KNN base models.

KNN is a distance-based classification model capable of handling both binary and multi-class attack classification. It is an instance-based learner who does less work during the training and more work during attacks' classification. It classifies the current network instance label as the majority attack type among the K closet instances. NB assumes that the features of the UNSW-NB15 dataset are independent of each other. It evaluates the joint conditional probability for each instance of the dataset in conjunction with each attack type. It returns the attack with the highest probability as the attack label of the network instance under analytics. C4.5 DT represents a classification problem as a tree with nodes corresponding to the UNSW-NB15 dataset features, and each leaf node corresponds to the attack type. It calculates the Gain Ratio of all the training dataset attributes by dividing the attribute's information gain with its split value. The Split value of an attribute is chosen by taking the average of all the values in the current attribute domain. The attribute with the highest gain ratio is selected as the root attribute. The root attribute divides the attributes into two branches. This procedure is repeated for each of the branches and all other subsequent branches until the tree is fully built. A new network instance is classified as the attack label that satisfied the values of its attributes.

MLR is a non-linear approach used for the building of linear model trees. It employs the use of linear regression to perform classification. For a classification problem with m class values: {c_1, c_2… cm}, m linear regression equations (LRm) are formulated. When an instance x is presented for classification, LR_m (x) is calculated for all m, and the class k with maximum LR_k(x) is predicted and returned. MMT is a model of trees induced by the M5 'algorithm, with linear regression at the leaves. It has been adapted to learn from and combine base-model predictions and allows an improved cyber-attack detection efficiency. The base model predictions are translated into a functional approximation for each attack type of the model trees. Given a classification problem with M multi-class, derived datasets for each Mattacktype were built from base model predictions. A regression function is generated for each of the M derived datasets for each form of attacks. Each of the M regression functions induces a new collection of base model predictions X to be improved. The regression function of the attack-type with the highest value is predicted and returned. MDTs is a type of decision tree used for combining several base classifiers predictions; its leaves nodes specify the base classifier that gives improved detection accuracy, the Multi-class detection predictions of each of the base classifier predicted over the possible attack class probability forms the MDT attributes. Given N set of base classifiers, M = (M_1, M_2... M_N), MDT assumes that each base classifier predicts a probability distribution over the attack's possible labels.

2.3 Performance Evaluation Metrics

Performance evaluation metrics play significant roles in assessing the binary and multi-class predictive model's performances and determining the model's fitness for the predictive purpose. The confusion matrix, also known as the error matrix, is used in this work for the evaluation of the cyber-attack detection models. Figure 2 shows the binary confusion matrix. It reflects four possible ways a sample point can be observed and classified;

- **a**:- number of times predicted traffic agreed with the observed class
- **b**:- number of times traffic was predicted as a class when it was observed to be another class
- **c**:- number of times traffic was not predicted as a class when it was observed to be the class
- **d**:- number of times predicted traffic was not observed to be the class

	Actual Attacks		
Predicted Attacks		+	-
	+	**a**	**b**
	-	**c**	**d**

Figure 2: Binary Confusion Matrix

This work used a multi-class confusion matrix to evaluate the performance and fitness of the cyber-attack detection models. Figure 3a shows the multi-class confusion matrix; its overall four possible ways of representing sample points all (observed and Actual) are shown in figure 3b and explain below;

	Classified			
Actual	n_{11}	N_{12}	…	n_{1k}
	n_{21}	n_{22}	…	n_{2k}
	…	…	…	…

	n_{x1}	n_{x2}	…	n_{xk}

Figure 3a: Multi-Class Confusion Matrix

		Actual Attacks	
Predicted Attacks		+	-
	+	**a** =$\sum_{i=1}^{k} n_{ii}$	**b** = $\sum_{i=1}^{k}\sum_{j\neq 1}^{k} nij$
	-	**c** = $\sum_{j=1}^{k}\sum_{i\neq j}^{k} nij$	**d** = $\sum_{i=1}^{k}\sum_{j\neq 1}^{k} nij$

Figure 3b: Overall Multi-class Confusion Matrix

a: is the number of positive instances correctly classified, which the sum of diagonal elements in a multi-class confusion matrix

b: is the number of positive instances that are incorrectly classified, which is the sum of non-diagonal elements in a multi-class confusion matrix

c: is the number of negative instances that are incorrectly classified, which is the sum of non-diagonal elements in a multi-class confusion matrix

d: is the number of negative instances that are correctly classified, which is the sum of non-row/column elements in a multi-class confusion matrix

Note: b = c

The popular binary performance evaluation metrics such as accuracy, specificity, and sensitivity do not give the true picture of the classifier's performance for multi-class problems. They assume that all the classes are evenly distributed and show biasness to network instances with the majority class. Mattew's correlation Coefficient (MCC) was used to evaluate the stacked ensemble detection models. MCC measures the coefficient of correlation between the predicted class and the observed class. The closer the coefficient to +1, the better the predictive model. A coefficient of +1 indicates a perfect correlation between model predictions and the actual class, and a coefficient of -1 indicates complete disagreement between the model predictions and the actual class.

The overall multi-class accuracy of a model is given by the ratio of the sum of diagonal elements to the total numbers of network instances presented for classification, as shown in equation 8. Accuracy of 1 implies a perfect model, and 0 indicates an imperfect model.

$$Overall\ Acc = \frac{a}{N} \quad = \quad \frac{\sum_{i=1}^{k} n_{ii}}{N} \tag{8}$$

The overall multi-class sensitivity/Recall, given in equation 9, is the model's ability to classify a network instance correctly as its actual value.

$$Overall\ Sensitivity/Recall = \frac{a}{a+c} = \frac{\sum_{i=1}^{k} n_{ii}}{\sum_{i=1}^{k} n_{ii} + \sum_{j=1}^{k} \sum_{i\neq j}^{k} nij} \tag{9}$$

The overall multi-class specificity, given in equation 10, is the model's capability to keep away from misclassifying a network instance if they are not. It is identical to overall accuracy.

$$Overall\ Specificity = \frac{d}{b+d} = \frac{\sum_{i=1}^{k} \sum_{i\neq j}^{k} \sum_{j=i}^{k} n_{ij}}{\sum_{i=1}^{k} \sum_{j\neq 1}^{k} nij + \sum_{i=1}^{k} \sum_{i\neq j}^{k} \sum_{j=i}^{k} n_{ij}} \tag{10}$$

The overall multi-class Precision, given in equation 11, measures a model's ability to correctly predict positive network instances.

$$Overall\ Precision = \frac{a}{a+b} = \frac{\sum_{i=1}^{k} n_{ii}}{\sum_{i=1}^{k} n_{ii} + \sum_{i=1}^{k} \sum_{j\neq 1}^{k} nij} \tag{11}$$

The F1-Score is a good metric for the evaluation of imbalanced data [30]; it represents the harmonic average of sensitivity and precision equation 12

$$F1 - Score = \frac{2 * (Overall\ sensitivity * Overall\ Precision)}{(Overall\ sensitivity + Overall\ Precision)} \tag{12}$$

MCC is defined in equation 13

$$MCC = \frac{((a*d)-(c-b))}{\sqrt{((a+c)(a+b)(d+c)(d+b)}} \ = \frac{((a*d)-(c^2))}{((a+c)(d+c))} \tag{13}$$

Since c = b,

$$MCC = \frac{((a*d)-(c^2))}{((a+c)(d+c))} \tag{14}$$

3.1 Results and Discussion

All our base and stacked ensembles cyber-attacks detection models were implemented using Python programming language and deployed on HP Core i5 CPU with a clocking rate of 2.4GHz, 64-bits BUS, 6GB RAM, 500GB Hard disk running on a Windows 7 Operating System. The information gain, Correlation, and Consistency Filter-based feature selection methods selected 32, 33, and 39 Multi-class detection features, respectively, as shown in Table 2. Table 3 shows the Naïve' Bayes base model's confusion matrix with the UNSW-NB15 dataset's reduced attributes. Table 4 shows the computed evaluation metrics for table 3. The base and metal models' confusion matrix was used to compute their corresponding computed performance evaluation metrics.

Table 2: Multi-Class Features Selection by the three Filtered Feature Selection Methods

Consistency Selected features (39)	Information gain Selected Features (32)	Correlation Selected Features (33)
dur, proto, service, state, spkts, dpkts, sbytes, dbytes, rate, sttl, sload, dload, sloss, dloss, sinpkt, dinpkt, sjit, djit, stcpb, dtcpb, dwin, tcprtt, synack, ackdat, smean, dmean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, is_ftp_login, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst, is_sm_ips_ports	dur, proto, service, state, spkts, dpkts, sbytes, dbytes, rate, sttl, dttl, sload, dload, sloss, dloss, sinpkt, dsinpkt, sjit, djit, tcprtt, synack, ackdat, smean, dmean, trans_depth, response_body_len, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst	dur, proto, service, state, spkts, dpkts, sbytes, dbytes, rate, sttl, dttl, sload, dload, sloss, dloss, sinpkt, dsinpkt, sjit, djit, swin, stcpb, dtcpb, dwin, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, is_ftp_login, ct_ftp_cmd, ct_flw_http_mthd, ct_src_ltm, ct_srv_dst, is_sm_ips_ports

Table 3: Confusion Matrix of Naive Bayes Model of the Correlation Reduced Dataset

classified as	**a**	**B**	**c**	**d**	**e**	**F**	**G**	**h**	**I**	**J**
a = analysis (2000)	1347	0	18	433	0	1	132	19	44	6
b = backdoor(1746)	235	1065	23	37	7	3	59	28	222	67
c = dos (12264)	645	3	8151	997	39	47	576	319	919	568
d = exploits (33393)	6330	6	381	17004	175	89	1915	727	2923	3843
e = fuzz/ers (18184)	3549	48	267	68	6731	657	523	1477	3993	871
f = generic (4000)	261	0	11	178	8	38105	97	1041	160	139
g = normal (56000)	3153	102	82	4301	2858	875	37671	1741	4856	361
h = reconnaissance (10491)	846	0	101	35	124	66	210	5845	1738	1526
i = shellcode (1133)	40	0	9	1	16	8	19	1	1013	26
j = worms (130)	0	0	1	6	0	0	6	8	15	94

Table 4: Computed Evaluation Metrics of the Naive Bayes Model of the Correlation Reduced Dataset

		Multi-Class Performance Metrics				***Overall Performance Metrics***	
Model	***Attack Classes***	***Accuracy***	***Recall***	***Precision***	***F1-Score***	***Accuracy***	***MCC***
Naïve Bayes Model of Correlation Reduced Features	Analysis	67.35%	0.6735	0.0821	0.0000	*66.74%*	*0.6305*
	Backdoor	61.00%	0.6100	0.8701	0.7172		
	Dos	66.46%	0.6646	0.9013	0.7651		
	Exploits	50.92%	0.5092	0.7374	0.6024		
	Fuzzers	37.02%	0.3702	0.6759	0.4784		
	Generic	95.26%	0.9526	0.9562	0.9544		
	Normal	67.27%	0.6727	0.9142	0.7751		
	Reconnaissance	55.71%	0.5571	0.5216	0.5388		
	Shellcode	89.41%	0.8941	0.0638	0.1191		
	Worms	72.31%	0.7231	0.0125	0.0246		

Table 5a shows the base model detection accuracy of the multi-class evaluation of the attacks test dataset. Their multi-class detection accuracy identifies the quality of the base models. All the C4.5 Decision tree models of information gain reduced features recorded the highest multi-class detection accuracy except for Shellcode attacks detection. Naïve Bayes model of consistency reduced dataset recorded the highest Shellcode attacks detection accuracy of 90.29%. All the KNN models and the C4.5 Decision tree model of the correlation reduced features failed to detect all analysis attacks. They recorded a zero detection accuracy for analysis attacks. All the base models, except the C4.5 Decision tree model of correlation reduced dataset, recorded their highest detection accuracy in detecting generic attacks. The C4.5 Decision tree model of correlation reduced features recorded its highest detection accuracy of 98.20% in detecting normal network traffic.

The work in [31] shows that F1-Score, a Precision based metric for evaluating machine learning models, gives a better model performance measurement than the accuracy metric in evaluating imbalanced class distribution. Value of F1-Score ranges between zero (0) to one (1), an F1-Score of less than 0.5 is considered poor and indicates a poor precision and recall of the model. The closer its value to one (1), the better the model under consideration. Table 5b shows the evaluated F1-Score values for each of the base models. The following can be deduced from Table 5b; all the base models perform fairly to detect Normal, Dos, Exploits, Fuzzers, and Generic attacks. All the base models perform poorly at detecting analysis attacks. C4.5 Decision tree model of the information gain reduced dataset recorded the best performance across all attack classes.

Table 5a: Multi-Class Detection Accuracy of the Base Models Evaluation of the Attacks' Test Dataset.

	Models of Information Gain Reduce Features			Models of Consistency Reduce Features			Models of Correlation Reduce Features		
Network Attacks	NB	KNN	C 4.5 Decision Tree	NB	KNN	C 4.5 Decision Tree	NB	KNN	C 4.5 Decision Tree
Analysis	60.15%	0.00%	22.85%	62.55%	0.00%	22.70%	67.35%	0.00%	0.00%
Backdoor	58.25%	46.96%	65.41%	60.48%	48.63%	64.43%	61.00%	46.30%	45.25%
Dos	55.16%	72.67%	82.11%	58.70%	72.81%	82.05%	66.46%	74.51%	79.47%
Exploits	50.86%	64.74%	71.90%	50.14%	62.94%	71.50%	50.92%	65.82%	69.48%
Fuzzers	50.76%	72.79%	79.42%	66.14%	71.74%	78.54%	37.02%	71.63%	78.71%
Generic	92.63%	98.35%	98.60%	97.26%	98.30%	98.49%	95.26%	97.91%	97.93%
Normal	76.48%	95.40%	97.38%	69.33%	96.08%	97.03%	67.27%	95.36%	98.20%
Reconnaissance	56.11%	63.08%	75.19%	57.21%	58.95%	73.62%	55.71%	63.64%	61.85%
Shellcode	88.44%	34.77%	69.20%	90.29%	58.16%	70.26%	89.41%	54.37%	69.20%
Worms	40.00%	26.15%	67.69%	42.31%	44.52%	64.52%	72.31%	37.69%	56.15%

Table 5c presented Matthew's Correlation Coefficient (MCC) of all the base models. From Table 5c, Naive Bayes' models recorded the least quality of multi-class detection. C4.5 DT models performed more than the other two base models in terms of Matthew’s Correlation Coefficient (MCC) score. The closer the MCC value to one (1), the better the model under consideration.

Table 5b: F1 Score values of the Base Models Multi-class Evaluation of The Attacks’ Test Dataset

	Models of Information Gain Reduce Features			Models of Consistency Reduce Features			Models of Correlation Reduce Features		
Network Attacks	NB	KNN	C 4.5 Decision Tree	NB	KNN	C 4.5 Decision Tree	NB	KNN	C 4.5 Decision Tree
Analysis	0.3643	0.0000	0.372	0.4021	0.0000	0.3699	0.1464	0.0000	0.0000
Backdoor	0.158	0.6021	0.7771	0.1744	0.6289	0.7729	0.7172	0.6018	0.6072
Dos	0.668	0.5377	0.5965	0.7008	0.5369	0.5809	0.7651	0.5578	0.5601
Exploits	0.647	0.7189	0.768	0.591	0.7178	0.7657	0.6024	0.7346	0.7525
Fuzzers	0.5593	0.7896	0.8489	0.6772	0.784	0.8385	0.4784	0.7824	0.8456
Generic	0.9572	0.9574	0.98	0.9745	0.9489	0.9787	0.9544	0.9514	0.9651
Normal	0.8321	0.9252	0.9683	0.8051	0.9237	0.9668	0.7751	0.9234	0.9731
Reconnaissance	0.4453	0.6882	0.8129	0.4456	0.6515	0.8189	0.5388	0.6756	0.6757
Shellcode	0.1183	0.3996	0.7183	0.1369	0.5972	0.722	0.1191	0.6033	0.7431
Worms	0.0612	0.2482	0.6848	0.0755	0.3986	0.6437	0.0246	0.3784	0.6432

Table 5c: Matthew's Correlation Coefficient (MCC) of the Base Models Evaluation on the Attacks' Test Dataset.

	Models of Information Gain Reduce Features			**Models of Consistency Reduce Features**			**Models of Correlation Reduce Features**		
Network Attacks	**NB**	**KNN**	**C 4.5 Decision Tree**	**NB**	**KNN**	**C 4.5 Decision Tree**	**NB**	**KNN**	**C 4.5 Decision Tree**
Matthew's correlation coefficient (MCC)	0.6621	0.8039	0.8575	0.7020	0.8006	0.8529	0.6305	0.8262	0.8530

The Comparative Performance of the Base Models based on F1-Score Metric, shown in figure 5, justified the ability of feature selection techniques to improve machine learning models' performances. All the base models of the reduced features justified the reduced features' ability to improve and records better performances than the models of all features of the cyber-attacks detection dataset [12, 13, 14].

Table 6 shows the multi-class attack detection performances of the three stacked ensemble models. MDT recorded the highest detection accuracy in eight (8) attacks detection; MDT stacking of information gain base models recorded the highest detection accuracy in Analysis, Backdoor, Dos, Exploits, Normal and Reconnaissance attacks. It also recorded the highest detection accuracy in detecting Worms and Shellcode attacks with the stacking correlation and consistency base model predictions. MLR and MMT stacking of the consistency base model predictions recorded the highest detection accuracy in Generic and Fuzzer attacks.

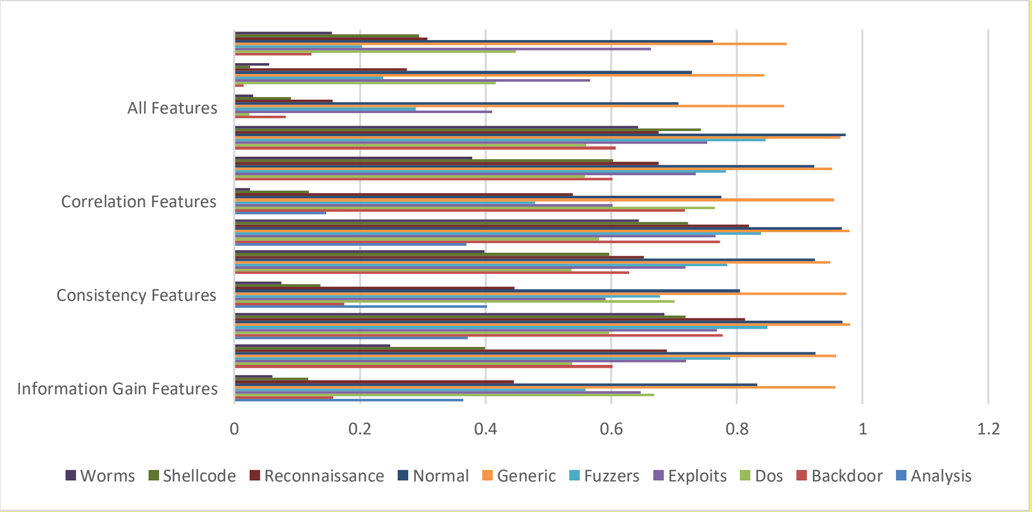

Figure 5: Comparative Performance of the Base Models based on F1-Score Metrics

Table 6: Multi-class Detection Accuracy of the Stacking of the Base Models Predictions with the Meta Learners

	MLR Stacking of Base Models Prediction			***MMT Stacking of Base Models Prediction***			***MDT Stacking of Base Models Prediction***		
Network Attacks	***Consistency Models. %***	***Information Gain Models %***	***Correlation Models %***	***Consistency Models %***	***Information Gain Models %***	***Correlation Models %***	***Consistency Models %***	***Information Gain Models %***	***Correlation Models %***
Analysis	80.55	82.25	79.95	84.70	89.75	84.15	87.55	90.10	85.75
Backdoor	83.39	86.08	79.38	86.43	87.92	83.73	87.06	90.26	85.28
Dos	92.21	95.22	92.97	97.37	97.55	94.32	97.78	97.95	95.42
Exploits	95.94	97.20	94.67	96.83	98.97	93.14	97.40	99.04	97.03
Fuzzers	94.98	95.03	96.82	98.77	98.66	94.87	97.23	98.17	97.94
Generic	99.35	98.85	98.14	98.85	98.98	98.48	98.99	99.14	98.62
Normal	99.13	99.08	98.93	99.10	99.11	98.20	99.27	99.41	98.93
Reconnaissance	94.57	95.54	93.28	96.43	96.47	92.2	97.78	98.49	96.59
Shellcode	91.11	91.35	90.56	91.62	94.17	94.00	94.97	93.65	92.32
Worms	85.38	84.62	83.85	85.38	86.15	86.15	86.15	86.92	87.69

MLR and MMT stacked ensemble recorded the least multi-class detection accuracy in six (6) and four (4) attacks, respectively; MLR stacking of the correlation base models recorded the least multi-class detection accuracy in Analysis, Backdoor, Generic and Worms attacks, MLR stacking of the consistency base models also recorded least multi-class accuracy in Dos and Shellcode attacks. MMT stacking of the correlation base model predictions recorded her least detection accuracy in Exploits, Fuzzer, Normal, and Reconnaissance.

Table 7 reports the cyber-attacks detection accuracy improvement of the base models. The details of the improvement of the base models' MCC by the stacked ensemble models are reported in Table 8.

Table 7: Stacked Ensemble Detection Accuracy Improvement of the Base Models

			Accuracy of the Stacked Ensemble Models			Stacked Ensemble Improvement of Base Models Accuracy		
		Accuracy (%)	MMT (%)	MDT (%)	MLR (%)	MMT (%)	MDT (%)	MLR (%)
Correlation	KNN	82.62	95.63	97.67	96.49	15.74	18.21	16.79
	C4.5 DT	85.30				12.11	14.51	13.12
	NB	66.74				43.28	46.34	44.57
Consistency	KNN	82.05	98.77	98.15	96.89	20.38	19.62	18.09
	C4.5 DT	86.77				13.84	13.12	11.67
	NB	70.20				40.71	39.82	38.03
Info Gain	KNN	82.35	98.48	98.75	97.39	19.59	19.91	18.26
	C4.5 DT	87.18				12.97	13.28	11.72
	NB	69.59				41.51	41.90	39.95

Table 8: Stacked Ensemble MCC Improvement of the Base Models

			MMC of the Stacked Ensemble Models			Stacked Ensemble Improvement of Base Models MCC Scores		
		MCC	MMT	MDT	MLR	MMT (%)	MDT (%)	MLR (%)
Correlation	KNN	0.8069	0.9514	0.9741	0.9610	18.52	21.05	19.59
	C4.5 DT	0.8366				14.30	16.74	15.33
	NB	0.6305				51.68	54.92	53.05
Consistency	KNN	0.8006	0.9770	0.9995	0.9654	23.38	22.60	21.03
	C4.5 DT	0.8529				15.80	15.07	13.59
	NB	0.66.8				47.67	46.75	44.86
Info Gain	KNN	0.8039	0.9831	0.9861	0.9710	22.50	22.84	21.15
	C4.5 DT	0.8575				14.84	15.16	13.57
	NB	0.6621				48.74	49.14	47.09

4. Conclusion

This work evaluates the performance of three selected meta-learning models for optimal multi-class detection of cyber-attacks in network packets. The study results show and confirm NB, C4.5 DT, and KNN ML suitability for solving a multi-class problem. It further affirms the knack of the duo of feature selection techniques and stacked ensemble learning to optimize ML models' performances. The information gain FS method gives better performance than the other two FS methods. C4.5 DT models return the best base model performance. The stacking of the predictions of the information gain base models with MDT gives the optimal cyber-attacks detection accuracy and MCC score. MDT stacked ensemble model of the predictions of the information gain reduced base models is therefore recommended for the optimal detection of the cyber-attacks in the UNSW-NB15 network intrusion dataset (packets).

5. Conflict of interest

The authors did not receive any financial aids or assistance from any individuals or institutions for this work. Further, we declared no conflicting interest that could undermine this research's reported work.

6. References

[1] Olasehinde O. O., Johnson O. V. & Olayemi O. C. (2020). Evaluation of Selected Meta-Learning Algorithms For The Prediction Improvement of Network Intrusion Detection System. 2020 International Conference in Mathematics, Computer Engineering, and Computer Science. (ICMCECS). doi:10.1109/icmcecs47690.2020.240893

[2] Olasehinde O. O., Alese B. K., & Adetunmbi A. O. (2018). A Machine Learning Approach for Information System Security. IJCSIS.16 (12). https://www.academia.edu/38339173/A_Machine_Learning_Approach_for_Information_System_Security

[3] Guenevere C.(2019) Toward realizing self-protecting healthcare Information systems: Design and security challenges, Chapter in a book: Advances in Computers, 114(113-149). https://doi.org/10.1016/bs.adcom.2019.02.003.

[4] Aryeh F. L., Alese B. K., Christian K. A. & Olasehinde O. O. (2020). ONDaSCA: On-demand Network Data Set Creation Application for Intrusion Detection System. International Journal of Computer Science and Information Security (IJCSIS), 18(5):111-115. https://sites.google.com/site/ijcsis/

[5] Delplace, A., Hermoso, S., & Anandita, K. (2020). Cyber Attack Detection thanks to Machine Learning Algorithms. ArXiv, abs/2001.06309.

[6] Kumar A, Glisson W, Cho H. (2020). Network Attack Detection Using an Unsupervised Machine Learning Algorithm. Hawaii International Conference on System Sciences. Doi:10.24251/HICSS.2020.795

[7] Blanco R., Malagón P., Cilla J. J. & Moya J. M.. (2018) Multiclass Network Attack Classifier Using CNN Tuned with Genetic Algorithms. 28th International Symposium on Power and Timing Modeling, Optimization, and Simulation (PATMOS), Platja (pp.177-182). doi: 10.1109/PATMOS.2018.8463997.

[8] Elmasry W., Akbulut A., & Halim Zaim A. (2019). An Empirical Study on Multi-class Classification-based Network Intrusion Detection. Journal of Computational Intelligence. 35(4): 919- 954. doi: 10.1111/coin.12220.

[9] Raogo Kabore, Yvon Kermarrec, & Philippe Lenca (2018). Performance Comparison For Multi-Class Classification Intrusion Detection In SCADA Systems Using Apache Spark. 8th Global Tech. Mining Conference, Leiden, Netherlands. ⟨hal-01876894⟩

[10] Aladesote I., Olutola A.,& Olasehinde O. Feature or Feature Extraction for Intrusion Detection System using Gain Ratio and Principal Component Analysis (PCA), Methodology. Communications on Applied Electronics (CAE), FCS. 4(3), DOI: 10.5120/cae2016652032.

[11] Mitra P., Murthy C. A., & Pal S. K. (2002) Unsupervised feature selection using feature similarity," IEEE Transactions on Pattern Analysis and Machine Intelligence. 24(3): 301–312.

[12] Almuallim H. & Dietterich T. G. (1994). Learning Boolean Concepts in the presence of many irrelevant features. Artificial Intelligence. 69(12): 279–305.

[13] Kaushik, S. (2016). Introduction to Feature Selection methods with an example (or how to select the right variables?). Analytic Vidya blog, 2016. (Accessed, 16th July 2020).https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with- an-example-or-how-to-select-the-right-variables/

[14] Brownlee, J. (2014). Feature Selection to Improve Accuracy and Decrease Training Time. (Accessed, 16th July 2020). https://machinelearningmastery.com/feature- selection-to-improve-accuracy-and-decrease- training- time/

[15] Yang, Y. & Pedersen J. O.(1997). A Comparative Study on Feature Selection in Text Categorization. In: Proc. of 14th International Conference on Machine Learning, ICML'97. (pp. 412-420).

[16] Jain, A.& Zongker D.(1997). Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. Pattern Anal. Mach. Intell. 19(2): 153-158.

[17] Cascaro R. J., Gerardo B. D. & Medina R. P.(2019). Filter Selection Methods for Multi-class Classification. Proceedings of the 2nd International Conference on Computing and Big Data. (pp.27-31). Doi: https://doi.org/10.1145/3366650.3366655

[18] Kunal, M. Dua. (2020) Attribute Selection and Ensemble Classifier based Novel Approach to Intrusion Detection System. International Conference on Computational Intelligence and Data Science (ICCIDS) Procedia Computer Science 167: 2191–2199.

[19] Al-Yaseen, W.L.(2020). Improving Intrusion Detection System by Developing Feature Selection Model Based on the Firefly Algorithm and Support Vector Machine. IAENG Int. J. Comput. Sci. 2019, 46(4): 534–540

[20] Zhou Y., Cheng G., Jiang S., & Dai M. (2020). Building an Efficient Intrusion Detection System based on Feature Selection and Ensemble Classifier. Computer Network. 174,(389–403) **doi**:10.1016/j.eswa.2017.08.002.

[21] Wu S. X., & Banzhaf W. (2010). The Use of Computational Intelligence in IDS, Journal of Applied Soft Computing 10(1) 1-35.

[22] Iwendi, C., Khan, S., Anajemba, J. H., Mittal, M., Alenezi, M., & Alazab, M. (2020). The Use of Ensemble Models for Multiple Class and Binary Class Classification for Improving Intrusion Detection Systems. Sensors (Basel, Switzerland). 20(9), 2559. https://doi.org/10.3390/s20092559

[23] Rajadurai, H., Gandhi, U.(2020). A stacked ensemble learning model for intrusion detection in a wireless network. Neural Computer and application. https://doi.org/10.1007/s00521-020-04986-5.

[24] Rajagopal, S., Kundapur, P. P., & Hareesha, K. S. (2020). A Stacking Ensemble for Network Intrusion Detection Using Heterogeneous Datasets. Security and Communication Networks**.** [4586875]. https://doi.org/10.1155/2020/4586875

[25] Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J. & Alazab, A.(2020). Hybrid Intrusion Detection System Based on the Stacking Ensemble of C5 Decision Tree Classifier and One-Class Support Vector Machine. Electronics. (9)173..

[26] Olasehinde O. O., Olayemi O. C. & Alese B. K.(2019). Multiple Model Tree Meta Algorithms improvement of IDS. Prediction Accuracy. 9(3) 891-897. doi: 10.20533.ijisr.2042.2019.0102.

[27] Moustafa N. & Slay j. (2015). UNSW-NB15: A Comprehensive DataSet for Network Intrusion Detection Systems Military Communications and Information Systems Conference. (pp. 1-7)

[28] Garcia-Teodoro, Diaz-Verdeio J., MaciaFernandez G. & Vazquz E. (2009). Anomaly-based Network Intrusion Detection: Techniques, Systems, and Challenges. Journal of Computers and Security. 28(1-2): 18-28.

[29] Olasehinde O. O**.,** Alese B. K., Adetunmbi A. O. (2018), Performance Evaluation of Bayesian Classifier on Filter-Based Feature Selection Techniques, International Journal of Computer Science and Telecommunications, 9(7):24-30

[30] Jeni L.A, Cohn J.F., De La Torre F. (2013). Facing imbalanced data–recommendations for the use of performance metrics, in 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE, 2013, pp. 245–251.

[31] Siblini W., Fréry J., He-Guelton L., Oblé F., Wang YQ. (2020) Master Your Metrics with Calibration. In: Berthold M., Feelders A., Krempl G. (eds) Advances in Intelligent Data Analysis XVIII. IDA 2020. Lecture Notes in Computer Science, vol 12080. Springer, Cham. https://doi.org/10.1007/978-3-030-44584-3_36

BIOGRAPHICAL NOTES

Olasehinde Olayemi O. PhD. is a Cyber Security Research Fellow and a lecturer with the Computer Science Department, Federal Polytechnic, Ile Oluji, Ondo State, Nigeria. He obtained his Bachelor of Technology, Master of Technology, and Doctor of Philosophy in Computer Science, from the Department of Computer Science, the Federal University of Technology, Akure, Ondo State, Nigeria, in 1995, 2012, and 2018 respectively. He is a member of the Nigeria Computer Society (NCS), Computer Registration Council of Nigeria (CPN). Cyber Corps of African. His research interests include cybersecurity, Computer Security, Data and Text Mining, Machine Learning, Bio-informatics, and Big Data Analytics.

Boniface Kayode Alese, Ph.D., is a Professor of Information and Cyber Security in the Department of Cyber Security, The Federal University of Technology, Akure, Nigeria. He holds a Ph.D. degree in Computer Science with a specialization in Information and Cyber Security from The Federal University of Technology, Akure, in 2004. He joined the Federal University of Technology services, Akure, in 1998 as a Graduate Assistant and rose to a Professor's position in 2014. He is a registered Information Technology practitioner. He is also a member of the Nigeria Computer

Society (NCS), Institute of Electrical and Electronic Engineering (IEEE), Computer Society, Association for Computing Machinery (ACM).***Information Systems Audit and Control Association (ISACA)**,* Information Systems Security Association (ISSA), and Cyber Security Experts Association of Nigeria (CSEAN) among members. He is widely traveled with almost 21 years of experience in research, training, and development. He was the Chair occupant of The First Bank of Nigeria endowment in Computer Science between 2012 and 2016. He has successfully supervised 18 Ph.D. students as a Major Supervisor. He has over 200 publications in reputable journals and referred conference proceedings.

Adetunmbi Adebayo is a Professor in the Department of Computer Science, the Federal University of Technology Akure, where he obtained his Ph.D. in Computer Science in 2008. He was a recipient of CAS-TWAS postgraduate fellowship at the Institute of Computing Technology, Beijing in 2006 and a Visiting Scholar to Massachusetts Institute of Technology in 2012 under MIT International Science and Technology Initiatives Empowering the Teachers Program. He has several publications in reputable peer-reviewed journals and has also served as a reviewer to several peer-reviewed journals. His research interests are information security, machine learning, and computational linguistics. He is a member of the IEEE computer society and Computer professional Council of Nigeria.

Aladesote Olomi Isaiah is a Ph.D. student in the Faculty of Computer Science and Information Technology, University Putra Malaysia, and a lecturer with the Computer Science Department, Federal Polytechnic, Ile Oluji, Ondo State, Nigeria. He completed his Master of Technology in the Federal University of Technology, Akure, Ondo State, Nigeria, in 2014. His research interests include cybersecurity, Computer Security, and Software Defined Network.

REFERENCE

Reference to this paper should be made as follows: Olasehinde O.O., Alese B. K., Adetunmbi A. O. & Aladesote O. I. (2020). Evaluation of Selected Stacked Ensemble Models for Optimal Multi-class Attacks Detection. *International Journal on Cyber Situational Awareness*, Vol. 5, No. 1, pp26-48