ISSN 2278-3091
Volume 10, No.4, July - August 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse161042021.pdf
https://doi.org/10.30534/ijatcse/2021/161042021

Comparison of Single and Ensemble Intrusion Detection Techniques using Multiple Datasets

Hassan Adegbola Afolabi[1], Abdurazzag Aburas[2]
[1]School of Electrical, Electronic and Computer Engineering
University of Kwazulu-Natal South Africa, 219048801@stu.ukzn.ac.za
[2]School of Electrical, Electronic and Computer Engineering
University of Kwazulu-Natal South Africa,aburasa@ukzn.ac.za

ABSTRACT

The advancement of Internet of Things (IoT) technology raises numerous security concerns, as new threats emerge every day. Prior to preventing these threats, they must be detected. This makes intrusion detection a major priority. However, datasets play a significant role in intrusion detection. The dataset used to evaluate machine learning-based solutions has an effect on their accuracy. Most of the time, these datasets do not accurately reflect real network traffic and contains lots of redundant and irrelevant features that undermine Intrusion Detection System (IDS) efficiency. Motivated by the above, our work focuses on extracting the most relevant features from four datasets namely CICIDS2017, IoTID20, NSL-KDD and N-BaIoT datasets using information gain approach. Then we evaluated and compared some single and ensemble classifiers based four important performance metrics. Finally, these algorithms were combined in an ensemble learner to see how well they performed. Our findings are considered to be relevant in the combination of strong classification algorithms in the development of IDS systems and experimental results indicates that feature selection can yield better accuracy.

Key words: Ensemble techniques, Feature selection, Internet of things, Intrusion detection systems, Machine learning.

1. INTRODUCTION

The increased use of Internet has greatly increased the data growth rate from different devices and created numerous security concerns. Various technologies like user authentication, data encryption and firewall have been used to address these security concerns. Though these countermeasures may prevent many kinds of attacks, they cannot quickly detect intrusion nor perform a thorough analysis of packets. Big data analysis and techniques are employed in order to handle intrusions more effectively due to high speed and enormous volume of data, the rapid development of sophisticated attacks and zero-day vulnerabilities on computer networks. Intrusion detection has become a major area of concern. They were developed to exclusively monitor computer networks, detect intrusions, attacks, unauthorized or any other malicious activities [1-3]. This will enhance security and compliment the shortcomings of other traditional security techniques.Intrusion detection systems can be classified into three types based on their detection method namely: anomaly-baseddetection, knowledge-based detection, and hybrid detection.

The knowledge based which is also known as signature-based technique rely on a database that contains signatures of existing attacks to look for a defined pattern. To keep up with emerging attacks, the database must be updated frequently. Therefore, only well-known attacks can be detected by this technique. On the other hand, anomaly-based IDSs also known as behavior-based because they monitor the system's, users', and network's normal behavior and warn the administrator if any deviation occurs. The ability of anomaly-based IDSs to detect novel threats is due to this feature. The hybrid-based detection system refers to a system that combines anomaly-based and knowledge-based intrusion detection.

An IDS's performance is greatly dependent on the datasets used to test and analyze it. In order to evaluate and test new approaches, appropriate and valid datasets are needed. Many researchers find this difficult which makes it a major task. While the majority of their tested with obsolete datasets [4], current network traffic data should be used to test IDS in order to make detection system more resilient [5]. However, implementing an efficient intrusion detection system could be a difficult task, given the abundance of redundant and irrelevant features in the dataset. It is tough to monitor all the features in the dataset, this could cause computational complexity

and decreased efficiency. As a result of this, in order to improve IDS detection accuracy, selected features from a dataset should be extracted prior to using any detection approach. A preprocessing technique known as feature selection has been shown to be a suitable solution for an IDS [6,7]. It discovers highly important features and removes unnecessary ones.

Motivated by the discussion above, our study will focus on the performance of various machine learning techniques used in detection classification systems when applied to four publicly available recent datasets.

The contributions of this work are as follows:

- We present an overview of intrusion detection systems that employ machine learning techniques.
- A feature extraction technique known as Information gain to extract the best feature was employed to manage large amounts of irrelevant features in the datasets.
- Five algorithms were evaluated, the majority of which fall under the category of individual and ensemble classifiers.
- We suggested a novel approach for intrusion detection that combines the benefits of feature selection, single and ensemble classifiers.
- We studied the performance of our approach and each analyzed classifier using four real traffic datasets.A Comprehensive comparison was done.

The rest of the paper is organized as follows: In Section II, we focus on some of the major related works in the area of intrusion detection. Section III describes the experimentation procedure, tools and methodology used in different steps of the evaluation. Our ensemble model is described in Section IV and Section V discusses the results of the experiments. The conclusion and future work are presented in sections VI and VII respectively.

2. RELATED RESEARCH WORK

Intruders update themselves and the tools they use to develop new cyber-attacks on a daily basis. Due to this, Intrusion detection techniques are being designed at a rapid pace to ensure that network systems are effectively secured against newly developed malware. Numerous researches have been conducted for this reason, and new ones are conducted daily to improve the efficacy of IDS systems. Research findings in a study conducted in [8] concludes that datasets representing exact network systems are now becoming more important to evaluate intrusion detection algorithms.

As a result, Mahoney et al. [9] studied and discovered that the DARPA/MIT Lincoln laboratory evaluation dataset results in an overly optimistic detection of network abnormality. Additionally, the authors proposed that this issue could be avoided by the combination of real-world traffic and simulated dataset. Later, an approach using random forest for misuse-based, anomaly-based and hybrid IDSs was presented by authors in [10]. Numerous machine learning techniques with improved accuracy have been developed over the past few years, a hybrid approach suggested in [11] which combines K-means clustering and the radial basis function (RBF) kernel of a support vector machine (SVM) is an example of such evolution. In addition to these advancements, various performance comparisons of these intrusion detection systems have been conducted. Belavagi et al. [12] used the NSL-KDD dataset to evaluate Logistic Regression, Gaussian Naive Bayes, Support Vector Machine, and Random Forest techniques. According to the author, Random Forest Classifier outperforms the other three algorithms. See table I below.

Table 1

ALGORITHMS	PRECISION (%)	ACCURACY (%)
Gaussian Naïve Bayes	79	79
Logistic Regression	83	84
Random Forest	76	75
Support Vector Machine	99	99

Additionally, Almseidin et al. [13] studied Random Forest, Random Tree, Bayes Network, Naïve Bayes, Decision Table, MLP and J48 machine learning algorithms in 2017. However, on the KDD dataset, decision tree has the lowest false negative value (0.002), but random forest outperforms in terms of accuracy. See table II below.

Table 2

ALGORITHMS	PRECISION (%)	ACCURACY (%)
Bayes Network	99.2	90.7
Decision Table	94.4	92.4
J48	98.9	93.1
MLP	97.8	91.9
Naïve Bayes	98.8	91.2
Random Forest	99.1	93.7
Random Tree	99.2	90.5

Likewise, Zaman et al. [14] conducted experiments to compare the precision, accuracy, and recall of Fuzzy C-Means, Radial Basis Function, k-Nearest Neighbors, Support Vector Machine, k-Means, Nave Bayes, and an ensemble technique combining all six algorithms. Kyoto+ dataset was used to evaluate these algorithms, and it was determined that Radial Basis Function outperformed the others. See the table below.

Table 3

ALGORITHMS	PRECISION (%)	ACCURACY (%)
Ensemble	88.4	96.7
Fuzzy C-Means	75	83.6
K means	75	83.6
K-Nearest Neighbors	95.6	97.5
Naïve Bayes	91.6	96.7
Radial Basis Function	92	97.5
Support Vector Machine	86.9	94.2

Also, in 2018, Aljawarneh et al. [15] presented a hybrid intrusion detection model using a voting scheme that combined Naive Bayes, J48, Random Tree, AdaBoostM1, Decision Tree, Decision Stump, and Meta Bagging. As a result, 99.81 percent detection accuracy was achieved.

Hajisalem et al [1] used an Artificial Bee Colony (ABC) and an Artificial Fish Swarm (AFS) to design a hybrid classification approach. They used Fuzzy C-Means Clustering (FCM) and Correlation-based Feature Selection (CFS) approaches to select features. This approach was applied to UNSW-NB15 and NSL-KDD datasets. 99% accuracy rate was obtained. The CSE-CIC-IDS-2017 dataset was developed by Sharafaldin et al. [4] since the current datasets did not fulfill today's demand for intrusion detection. A test environment was created with network attackers and victims to generate this dataset. Attacks such as distributed denial of service, denial of service, Infiltration attack, Web attack, brute force, botnet and heart bleed were organized in the test environment. Additionally, machine learning approaches were used to evaluate system performance. Ferrag et al [16], studied some deep learning algorithms namely deep neural networks (DNN), recurrent neural networks (RNN), convoluted neural networks (CNN), deep autoencoders (DA), deep belief networks (DBN), deep Boltzmann machines (DBM), and restricted Boltzmann machines (RBM) when implemented on CSE-CIC-IDS2018 and Bot-IoT datasets. The classification success of deep learning is then compared to the classification time for these data sets. Hassan et al. [17] evaluated ML classifiers such as Artificial Neural Network (ANN), Support Vector Machine (SVM), Gaussian Naïve Bayes (NB), Decision Tree (DT) and Random Forest (RF) with both KDD'99 and ISCX1DS2012 datasets. The SVM outperforms other algorithms for both datasets and NB algorithm was the least accurate for both datasets. Additionally, a standard deep neural network (DNN) approach was presented in [18] This approach was based on back propagation algorithm and trained using 3 hidden layers. When evaluated using an unlabeled CICIDS2017 dataset, an average accuracy of 84.5% was obtained.Authors in [19] presented a novel framework for intrusion detection using ensemble approach. This architecture evaluated various supervised and unsupervised machine learning algorithms on CICIDS-17 dataset. Experimentshows that ensemble approach provides better performance.

3. EXPERIMENTATION

This section briefly discusses fundamental concepts such as dataset description, pre-processing of the dataset, feature extraction procedure, machine and deep learning methods used, and model design. We examined two network datasets which reflect actual real-world network traffic namely: NSL-KDD and CICIDS 2017 and Two IoT datasets namely N-BaIoT and IoTID20. In order to obtain reliable results, the following steps was followed before the analysis of the algorithms:

- Import the datasets that will be used for algorithm training and testing into the Google Colaboratory environment.
- Preprocess the data to select relevant attributes. Mutual Information Feature selection (Information Gain) was applied on all four selected datasets to extract 20 most relevant attributes.
- Applied Feature engineering to create columns for each attack class to enable us obtain results in multiclass.
- Analyzed the selected algorithms individually with the preprocessed data attributes and obtain results to check individual performance.
- Combination of 3 other algorithms with DNN as our base classifier (algorithms combined are (SGD, LGBM, XGBOOST AND DNN).
- The output of the four base classifiers was used to train the ensemble model (Meta classifier which uses LR algorithm).
- Test was done with a different portion of the dataset on the meta classifier to obtain experimental results based on the following metrics (Section III.D): true positive, false positive, precision, recall and F- measure.

Note:

a) A comparison was done with the performance of each individual algorithm so as to find the best possible algorithm to combine with the DNN to build the stacking model.
b) The choice of using Logistic Regression algorithm as the metaclassifier is because it's the weakest as per our results with all other algorithms.

The experiment is performed on Google Colaboratory under python 3 using TensorFlow and Graphics Processing Unit (GPU). Installed on Mac OS X.

3.1 Dataset Description

As discussed in the previous sections, this study considered four datasets for evaluation purpose, CICIDS2017, NSL-KDD, N-BaIoT 2018 and IoTID20 datasets.

A. CICIDS2017

The Canadian Institute for Cybersecurity generated this dataset in 2017. The dataset was created by setting up a victim and an attacker network laboratory environment. The CICIDS2017 is a publicly accessible dataset that resembles real-world IDS network traffic [20]. Contained in this dataset are benign or 'normal' traffic and seven common and most recent attacks that simulates real-world data [4]. The attacks contained in this dataset include Brute Force, Distributed Denial of Service (DDoS), Web, Infiltration, Botnet, and Port Scan attacks. Over 2 million records and 78 features are contained in CICIDS2017.

B. NSL-KDD

NSL-KDD is a freely available dataset that was created to address the shortcomings in the KDD 99 dataset [21–23]. A benefit of this dataset is that it has no insignificant records in the train set, which means that the classifiers will not be biased towards more repeated records. According to [8], this dataset still lacks public network data.The NSL-KDD dataset consists of 42 attributes. This dataset contains denial of service (DoS), remote to local (R2L), user to root (U2R), and probe attacks.

C. N-BaIoT

N-BaIoT Dataset was created in 2018 to resolve the inadequacy of publicly available botnet datasets, particularly for IoT. Created using real traffic data from nine commercial IoT devices infected by authentic botnets from two families, Mirai and BASHLITE, which are two of the most prevalent IoT-based botnets that have already demonstrated their malicious capabilities. [24-25]. The aim was to use anomaly detection algorithms to differentiate between benign and malicious traffic data. However, the dataset could also be useful for multi-class classification because the malicious data is classified into 10 attacks classes carried out by two botnets, plus one "benign" attack class. The N-BaIoT dataset contains 115 independent features in each file, as well as a class label generated from the respective file name (e.g., "benign" or "TCP attack").

D. IoTID20

IoTID20 dataset is a new dataset proposed by [26]. Originally created using two basic smart home devices, the SKT NUGU (NU100) and the EZVIZ Wi-Fi Camera (C2C Mini O Plus 1080P) [27]. All devices involved was connected to the same wireless network. The dataset consists of 42 raw network packet files (pcap) collected using the wireless network adapter's monitor mode at various time points. The new dataset, on the other hand, contains a more extensive network and flow-based features. These flow-based features could be used in the analysis and evaluation of a flow-based Intrusion detection system. The packet files description in this dataset include 'benign' as normal traffic, and 'attack traffic' for attack classes such as MITM ARP spoofing, DoS, Scan and mirai.

3.2 Selection of Features

The selection of features (attributes/variables) is an important step to develop an intrusion detection model that is effective. Given the huge number of irrelevant features in network data, it is important to extract only the necessary attributes to minimize processing time and achieve a higher detection rate and accuracy. The method of feature (attribute) selection is an important data preprocessing approach that is used to extract a subset of relevant features (variables/attributes) in order to improve the performance of learning algorithms. Additionally, this process reduces the amount of storage required.Feature selection methods can be classified into three categories such as embedded, filter, and wrapper methods. The Filter and Wrapper methods are the most frequently used [28]. In this study, we employed the mutual information gain feature selection approach.Information Gain is a single-attribute evaluator that is used in relation with the Ranker search method to score all attributes based on their information gain. This is used to evaluate the value of each attribute by calculating the information gain in relative to the class. The score is determined by how much information about the classes is obtained when that feature is used.The Information Gain equation is shown in Eq. 1,

$$IG(X)=H(Y)-H(Y|X) \quad (1)$$

where H(Y) and H(Y|X) are the entropy of Y and the conditional entropy of Y for given X, respectively [29]. For this research, only 20 attributes were selected from each dataset when considering threshold values 0.29, 0.40, 0.73 and 0.42 for datasets NSL-KDD, CIC2017, N-BaIoT and IoTID20 datasets respectively.Attributes whose information gain value is below the considered threshold value are removed from the dataset. Figure 1- 4 below shows the description of the selected attributes.

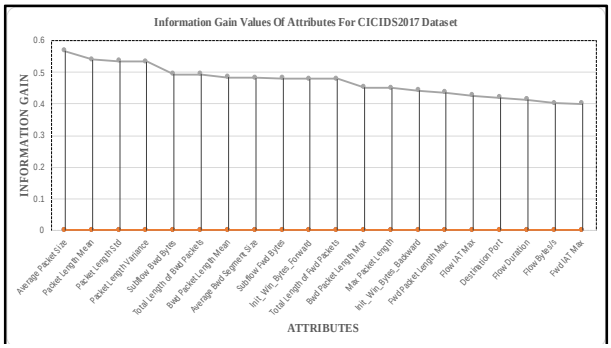

Figure1: CIC2017 attributes

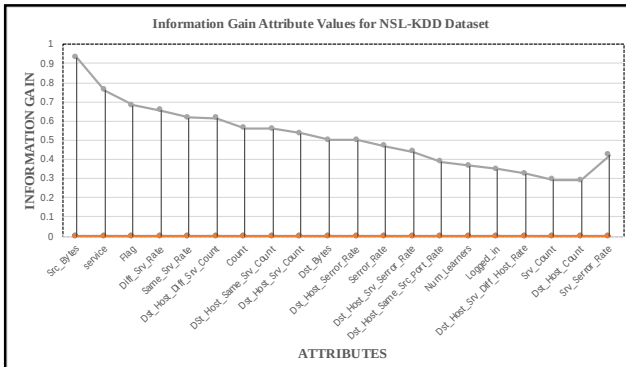

Figure 2: NSL-KDD attributes

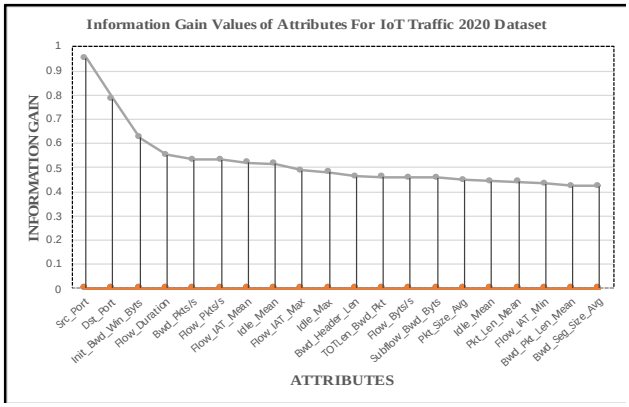

Figure3: IoTID20 attributes

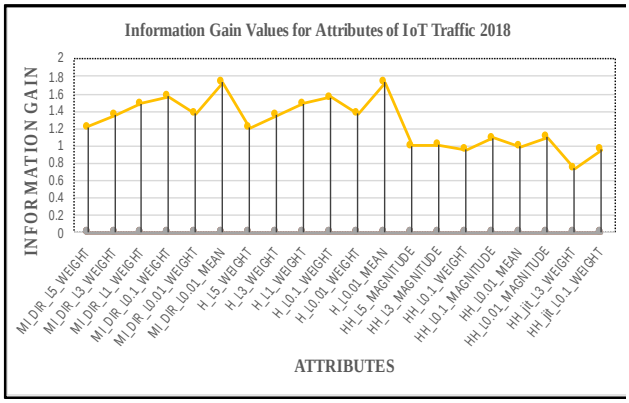

Figure 4: IoTID18 attributes

3.3 Evaluation of Algorithms

For evaluation purpose, this project has considered LR, SGD, LGBM, XGBOOST, and DNN algorithms.

A. Logistic Regression

Logistic Regression is a type of supervised machine learning method that is used to classify data. It can be used with categorical dependent variables. This algorithm has gained importance in recent years and its application has grown tremendously. The objective of the logistic regression algorithm is to assign data to their appropriate classes based on their correlation.For a mathematical expression of logistic regression, let us look at a simple linear regression equation below:

$$y = \beta_0 + \beta_1 * x \quad (2)$$

apply sigmoid function to the above equation will give:

$$p = \frac{1}{1+e^{-y}} (3)$$

Logistic regression formula can be derived by substituting eq. 2 in eq. 3 to give.

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots \beta_n x_n \quad (4)$$

It has a value between 0 and 1

$\beta_0, \beta_1 \ldots . \beta_n$ are the regression parameters,

$x_1, x_2 \ldots . x_n$ are the predictor values.

B. Stochastic Gradient Descent

Stochastic gradient descent (SGD) is a machine learning optimization approach that is frequently used to calculate the model parameters that best fit the expected and actual outputs. They are a variant of gradient descent techniques that's solves the issues of computational time. In SGD, the gradient of a randomly selected subset of the observations rather than all of them is calculated [30].

$$\nabla E(W_t, x_i, y_i) (5)$$

$W_t = Weight\ vector$, x_i= given data instances,

$y_i = output$, ∇E=true gradient.

C. Light Gradient Boosting Machine

LGBM is a high-performance gradient boosting system based on the decision tree approach that may be used for ranking, classification, and a variety of other machine learning tasks. LGBM splits the tree leaf wise based on the best fit. Thus, when growing on the same leaf in LGBM, the leaf-wise approach can reduce loss significantly more than other existing boosting techniques.A diagrammatic explanation is given in the figure below.

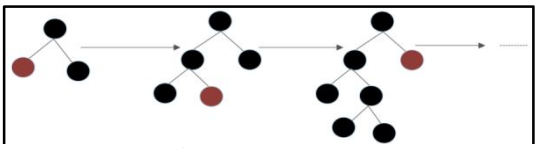

Figure 5: Leaf Wise Tree Growth in LGBM

D. Extreme Gradient Boosting

The XGBoost algorithm is a regression tree model classifier [31]. It provides parallel tree boosting (i.e. GBDT, GBM) that addresses a wide variety of data science issues quickly and accurately.

E. Deep Neural Network

A deep neural network (DNN) is a type of artificial neural network (ANN) that contains one or more layers between the input and output layers. A DNN is composed of the following basic components like neurons, synapses, weights, biases, and functions. It is composed of sequential linear functions and nonlinear activation functions and can be mathematically expressed as below:

$$y = \sigma(Wx + b) \quad (6)$$

Where y, W, x and b are outputs, weights, inputs and biases respectively. $\sigma(.)$is known as the activationfunction

3.4 Performance Metrics

We obtained the mean value for the following metrics during the performance analysis of all algorithms evaluated in the project: Accuracy, Precision, Recall, and F-score Accuracy, Precision, Recall, and F-score.

A. Accuracy

In classification problems, accuracy refers to the amount of accurate predictions divided all possible predictions. The mean accuracy rate is the average accuracy rate for each attack class in a given dataset.

$$Mean\ accuracy = \left(\frac{TP+TN}{TP+FP+FN+TN}\right)(7)$$

B. Precision

It is the ratio of accurate positive results to the number of predicted positive results by the algorithm.

$$Mean\ Precision = \left(\frac{TP}{TP+FP}\right)(8)$$

C. Recall

It is calculated by dividing the number of accurate positive results by the total number of relevant samples.

$$Mean\ Recall = \left(\frac{TP}{TP+FN}\right)(9)$$

D. F-score

This is the harmonic mean between recall and precision. It clearly shows how efficient an algorithm is. F-Score ranges between [0,1] and tries to find the balance between precision and recall.

$$Mean\ Fscore = \left(\frac{1}{2 * \frac{1}{\frac{1}{precision}+\frac{1}{recall}}}\right)(10)$$

Let's assume TRUE/FALSE to be 1/0, there are 4 important terms to note from the above metrics.

TP= True positive which simply means that both the prediction and actual output is YES (1)
TN= True negative which simply means that both the prediction and actual output is NO (0)
FN = False negative, means that the prediction NO (0) is different from the actual output YES (1).
FP= False Positive, means that the prediction YES (1) is different from the actual output NO (0)

4. PROPOSED ENSEMBLE APPROACH

Based on an extended experiment conducted in the previous sections, we present a stacking-based ensemble learning technique for an intrusion detection system. This model uses LR as a meta classifier and combines SGD, LGBM, XGBOOST, and DNN algorithms as base classifiers.Our proposed approach comprises two main stages namely:

Stage 1: involves the training of the base classifiers on each input dataset

Stage 2: Involves the training of the meta-classifier on the outputs of each individual base classifiers in the ensemble.

The framework of our proposed technique is shown in figure 6 below.

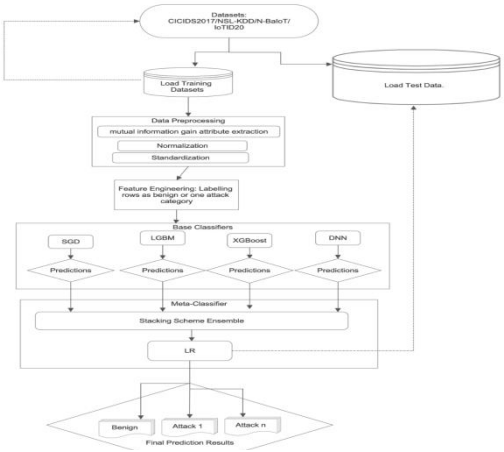

Figure 6: Framework of the Proposed Stacking Ensemble

An advantage of using this approach is that meta-classifier in the second stage can rectify the shortcomings of any or all of the base classifiers in the first stage. Since the objective is to obtain significantly better outcomes, our ensemble

technique must outperform the results of the best base classifier in the overall model.

1: Input: Training and Testing datasets D
2: Impose cross validation in order to prepare a training set for meta-classifier
3: Randomly split D into "x" equal size subsets $\{D_1, D_2, \ldots D_x\}$
for x← 1 *to X*
4: Feature selection to reduce numbers of feature applied using mutual information gain approach at threshold values 0.29, 0.40, 0.73, 0.42 for NSL-KDD, CIC2017, N-BaloT and IoTID20 datasets respectively.
5: Feature engineering to extract only relevant features from the selected features.
6: Train base classifiers namely DNN, SGD, LGBM and XGboost
for z← 1 *to Z*
Train a classifier T_{zx} from D or D_x
End for
7: Combine base algorithms using stacking scheme
8: Use Logistic Regression as meta classifier
9: Formulate a training set and extract new instances for meta-classifier
End for
End for
10: Returns results from ensemble

Algorithm 1: Stacking Ensemble Pseudo Code

5. RESULTS AND ANALYSIS

In this work, we compared the classification performance of a few machine learning techniques namely Logistic Regression, Stochastic Gradient Descent, Light Gradient Boosting Machine, and Extreme Gradient Boosting. Accuracy, precision, recall, and F-score are the metrics used in this work. The experiment was carried out on Google Colaboratory using Python 3, TensorFlow, and a Graphics Processing Unit (GPU) installed on a Mac OS X 2.8 GHz Intel Core i7 CPU, 16.00 GB RAM 2133MHz LPDDR3.The experiments are in phases. In the first phase: Information gain feature selection was applied to all the datasets considered for this work. 20 best attributes were extracted and the results were shown in figure 1-4.
In the second phase. A classification technique is applied on all datasets involved. Figures: 7(a-e) shows the overall performance of each technique relative to the corresponding datasets.

Figure 7(a): Outcomes of Logistic Regression on the Experimental Datasets

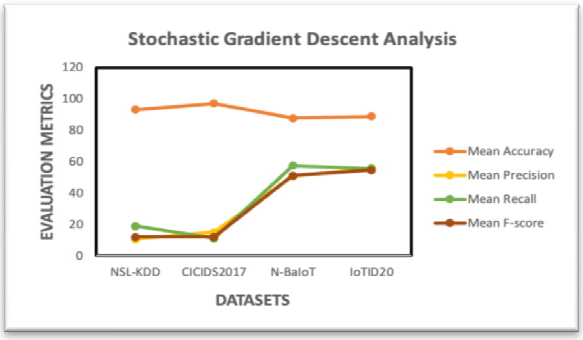

Figure 7(b): Outcomes of Stochastic Gradient Descent on the Experimental Datasets

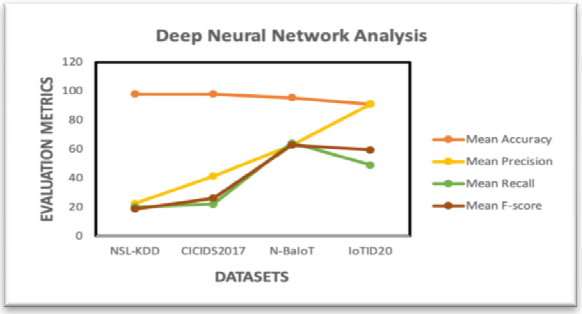

Figure 7(c): Outcomes of Deep Neural Network on the Experimental Datasets

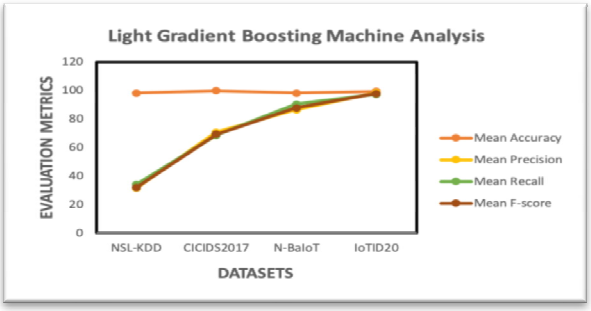

Figure 7(d): Outcomes of Light Gradient Boosting Machine on the Experimental Datasets

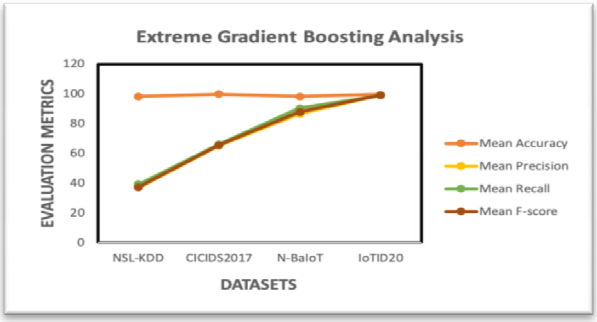

Figure 7(e): Outcomes of Extreme Gradient Boosting on the Experimental Datasets

From the results above, it shows that LGBM and XGBoost are the strongest learners in terms of accuracy and F-score. Since They give the best performance when applied to N-BaIoT and IoTID20

datasets, we chose to combine them with SGD and DNN as base classifiers to construct an ensemble stacking classifier and Logistic regression as meta classifiers. In order to evaluate the performance of our proposed stacking ensemble method, we compare all algorithms used in this work in terms of accuracy and F-score metrics as figures 8(a)-(b) below:

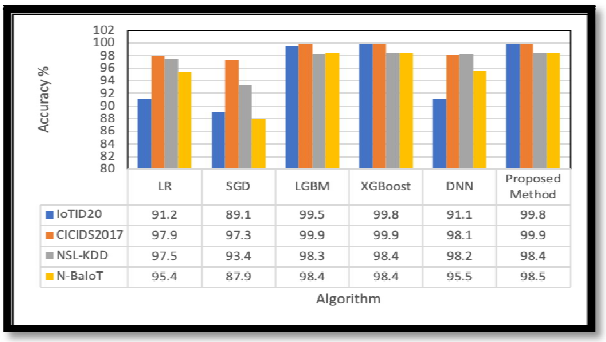

Figure 8(a): Performance comparison across all datasets based on accuracy

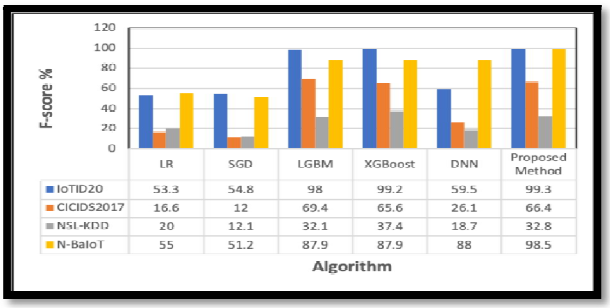

Figure 8(b): Performance comparison across all datasets based on F-score

The chart above shows that the LR algorithm when used as a meta classifier yields better accuracy and F-score than LR as a single classifier and amongst the other algorithms.

We observed that the performance of these algorithms depends on the kind of datasets employed because algorithms like LGBM, XGBoost and DNN gave better results with N-BaIoT and IoTID20 datasets. Regardless, our proposed method outperformed all other algorithms as shown in the tables below.

Table 4

Algorithm	Mean Accuracy (%)	Mean Recall (%)	Mean Precision (%)	Mean F-score (%)
LR	91.2	52.2	60.4	53.3
SGD	89.1	55.8	55.8	54.8
LGBM	99.5	97.6	98.5	98.0
XGBoost	99.8	99.0	99.5	99.2
DNN	91.1	49.1	91.1	59.5

The table 4 above shows the recorded values of accuracy, precision, recall and F-score when all algorithms are tested with IoTID20 dataset. It was observed that LGBM and XGBoost had the best performance having 99.5% and 99.8% accuracy, 98.5% and 99.5% precision, 97.7% and 99.0% recall, 98.0% and 99.2% F-score respectively. In contrast, SGD performed relatively poor in terms of accuracy and precision having 89.1% and 55.8% respectively. DNN had the least performance in terms of recall with 49.1% and LR yielded the least F-score of 53.3%.

Table 5

Algorithm	Mean Accuracy (%)	Mean Recall (%)	Mean Precision (%)	Mean F-score (%)
LR	97.9	15.7	19.2	16.6
SGD	97.3	11.2	15.1	12.0
LGBM	99.9	68.8	70.8	69.4
XGBoost	99.9	66.2	65.2	65.6
DNN	98.1	22.1	41.4	26.1

The next the experiment was conducted using CICIDS 2017 dataset. Table 5 above clearly shows the results obtained. Stochastic Gradient descent yielded the lowest performance with 97.3%,11.2%, 15.1% and 12% for accuracy, recall, precision and F-score respectively. Whereas, LGBM outperformed all the other algorithms, with accuracy, precision, recall and F-score obtained as 99.9%, 70.8%, 68.8% and 69.4% respectively.

Table 6

Algorithm	Mean Accuracy (%)	Mean Recall (%)	Mean Precision (%)	Mean F-score (%)
LR	97.5	21.5	20.9	20.0
SGD	93.4	18.9	10.7	12.1
LGBM	98.3	34.3	31.7	32.1
XGBoost	98.4	39.3	37.2	37.4
DNN	98.2	20.1	22.5	18.7

Table 7

Algorithm	Mean Accuracy (%)	Mean Recall (%)	Mean Precision (%)	Mean F-score (%)
LR	95.4	54.1	55.9	55.0
SGD	87.9	57.5	50.9	51.2
LGBM	98.4	90.6	86.5	87.9
XGBoost	98.4	90.6	86.6	87.9
DNN	95.5	64.4	62.8	62.8

Results obtained from the experiments using NSL-KDD and N-BaIoT are recorded in tables 6 and 7 above. We observed that SGD had the lowest performance with both dataset except for Logistic regression that the lowest recall of 54.1% with N-BaIoT dataset. Extreme Gradient Boosting algorithm yielded the highest results with (98.4%, 39.3%, 37.2%, 37.4%) and (98.4%, 90.6%, 86.6%, 87.9%) for accuracy, recall, precision and F-score respectively.

Table: 8

Linear Regression Stacking Ensemble (Proposed Method)	Mean Accuracy (%)	Mean Precision (%)	Mean Recall (%)	Mean F-score (%)
IoTID20	99.8	99.5	99.0	99.3
CICIDS'17	99.9	66.6	66.2	66.4
NSL-KDD	98.5	33.0	34.0	32.8
N-BaIoT	98.5	86.6	90.6	88.0

Considering our proposed stacking ensemble model, when it was evaluated using IoTID20 dataset, it achieved the highest F-score of 99.3% and accuracy, recall and precision similar to XGboost Thus, showing that our model was able to achieve the best results similar to the best algorithms for IoTID20 dataset as shown in Table 8. However, the proposed stacking ensemble model performed better than other algorithms considered for analysis in this work when evaluated with N-BaIoT dataset. Thus, it shows that the new technique can be used to deliver better results for Intrusion detection system in an IoT platform.

6. CONCLUSION AND FUTURE WORKS

While several machine learning approaches have been presented to improve the effectiveness of IDSs, Identification of relevant features in a dataset that has a substantial impact on IDS performance is a major challenge. Hence, with better feature selection, an efficient IDS can be designed. This work investigated the efficiency of the following techniques: LR, SGD, LGBM, XGBoost, and DNN. Evaluation was done using four datasets namely: CICIDS2017, IoTID20, NSL-KDD and N-BaIoT'18 datasets. 20 most relevant features from each dataset were extracted using the information gain feature extraction approach, and the techniques were compared based on four metrics: accuracy, precision, recall, and f-score. According to the experimental results, feature selection can enhance detection accuracy. Also, we presented an ensemble learning strategy based on the stacking of the analyzed algorithms. The experiment demonstrated that our ensemble technique outperforms other single classifiers across all datasets examined especially IoT traffic datasets.This demonstrates that our work has proven to be fairly significant in getting a better understanding of how to develop security solutions for IoT and our technique can be used for practical application of IDSs.Further research into more diverse base learners and alternative combination methods to improve these outcomes, comparing our technique with existing state of the art techniques is envisioned.

REFERENCES

[1]. V. Hajisalem, S. Babaie, A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection, Comput. Netw. 136 (2018) 37–50, https://doi.org/10.1016/j.comnet.2018.02.028.

[2]. Z. Inayat, A. Gani, N.B. Anuar, M.K. Khan, S. Anwar, Intrusion response systems: foundations, design, and challenges, J. Netw. Comput. Appl. 62 (2016),53–74, https://doi.org/10.1016/j.jnca.2015.12.006.

[3]. A.S. Ashoor, S. Gore, Difference between intrusion detection system (IDS) and intrusion prevention system (IPS), Commun. Comput. Inf. Sci. (2011) 497–501, https://doi.org/10.1007/978-3-642-22540-6_48.

[4]. I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, toward generating a new intrusion detection dataset and intrusion traffic characterization, in: ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv., 2018, pp. 108–116, https://doi.org/10.5220/ 0006639801080116.

[5]. M. Ahmed, A. N. Mahmood and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19-31, 2016.

[6]. H. Hota, A.K. Shrivas, Decision Tree Techniques Applied on Nsl-kdd Data and Its Comparison with Various Feature Selection Techniques, in: Advanced Computing, Networking and Informatics-Volume 1, Springer, 2014, pp. 205–211, doi:10.1007/978-3-319-07353-8_24.

[7]. C. Khammassi, S. Krichen, A ga-lr wrapper approach for feature selection in network intrusion detection, Comput. Secur. 70 (2017) 255–277, doi:10.1016/j.cose.2017.06.005.

[8]. J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," in *ACM Transactions on Information and System Security*, 2000.

[9]. M. Mahoney and P. Chan, "An analysis of the 1999 DARPA / Lincoln laboratory evaluation data for network anomaly detection," in *Recent Advances in Intrusion Detection, 6th International Symposium, RAID 2003, Pittsburgh, PA, USA*, 2003.

[10]. J. Zhang, M. Zulkernine and A. Haque, "Random-forests-based network intrusion detection systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649 - 659, 2008.

[11]. U. Ravale, N. Marathe and P. Padiya, "Feature selection-based hybrid anomaly intrusion detection system using K means and RBF kernel function," in *Proceeding of International Conference on Advanced Computing*, 2015.

[12]. M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," *Procedia Computer Science*, vol. 89, pp. 117-123, 2016.

[13]. M.Almseidin, M. Alzubi, S. Kovacs and M. Alkasassbeh, "Evaluation of Machine Learning Algorithms for Intrusion Detection System," in *15th International Symposium on Intelligent Systems and Informatics*, 2017.

[14]. M. Zaman and C. H. Lung, "Evaluation of Machine Learning Techniques for Network Intrusion Detection," in *IEEE/IFIP Network Operations and Management Symposium*, 2018.

[15]. S. Aljawarneh, M. Aldwairi and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152-160, 2018.

[16]. M.A. Ferrag, L. Maglaras, S. Moschoyiannis, H. Janicke, Deep learning for cyber security intrusion detection: approaches, datasets, and comparative study, J. Inf. Secur. Appl. (2020) 50, https://doi.org/10.1016/j.jisa.2019.102419.

[17]. H.A. Afolabi, A.A. Aburas, "An Evaluation of Machine Learning Classifiers for Prediction of Attacks to Secure Green IoT Infrastructure"*International Journal of Emerging Trends in Engineering Research*, 9(5), May 2021,549–557, https://doi.org/10.30534/ijeter/2021/03952021.

[18]. H.A. Afolabi, A.A. Aburas, "Proposed Back Propagation Deep Neural Network for Intrusion Detection in Internet of Things Fog Computing" *International Journal of Emerging Trends in Engineering Research*, 9(4), April 2021, 464 – 469,https://doi.org/10.30534/ijeter/2021/2394 2021.

[19]. S. R. Khonde, V. Ulagamuthalvi, "Hybrid Framework for Intrusion Detection System using Ensemble Approach" *International Journal of Advanced Trends in Computer Science and Engineering*, 9(4), July – August 2020, 4881 – 4890,https://doi.org/10.30534/ijatcse/2020/99 942020

[20]. Intrusion Detection Evaluation Dataset (CICIDS2017),"[Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. [Accessed 08 04 2019].

[21]. NSL-KDD dataset," [Online]. Available: https://www.unb.ca/cic/datasets/nsl.html. [Accessed 08 04 2019].

[22]. M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA, 2009, p. 2009, https://doi.org/10.1109/CISDA.2009.5356528

[23]. S. Revathi and A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion Detection," *International Journal of Engineering Research & Technology*, vol. 2, no. 12, 2013.

[24]. Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, A. Shabtai, and Y. Elovici "N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders", IEEE Pervasive Computing, Special Issue - Securing the IoT (July/Sep 2018).

[25]. Y. Mirsky, T. Doitshman, Y. Elovici& A. Shabtai 2018, "Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection", in Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA.

[26]. Ullah I., Mahmoud Q.H. (2020) A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In: Goutte C., Zhu X. (eds) Advances in Artificial Intelligence. Canadian AI 2020. Lecture Notes in Computer Science, vol 12109. Springer, Cham. https://doi.org/10.1007/978-3-030-47358-7_52

[27]. Hyunjae Kang, Dong Hyun Ahn, Gyung Min Lee, Jeong Do Yoo, Kyung Ho Park, Huy Kang Kim, September 27, 2019, "IoT network intrusion dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/q70p-q449.

[28]. Sheena, Krishan Kumar, Gulshan Kumar: Analysis of Feature selection Techniques: A Data Mining Approach, International Journal of Computer Applictions, ICAET 2016, IJCA2016 (1):17–21.

[29]. Zahra Karimi, Mohammad Mansour and Ali Harpunabadi: Feature Ranking in Intrusion Detection Dataset using Combination of Filtering Methods, IJCA, Vol 78, No 4, September 2013.

[30]. Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[31]. Chen, T., &Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acmsigkdd international conference on knowledge discovery and data mining* (pp. 785-794)