Distributed Intrusion Detection System for Cloud Environments Using Deep Learning Machine Algorithms

MSc Research Project
Cybersecurity

Oyindeinbofa Pibowei
Student ID: 20165765

School of Computing
National College of Ireland

Supervisor: Prof Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Oyindeinbofa Pibowei
Student ID:	X20165765
Programme:	M.Sc Cybersecurity **Year:** 2022
Module:	M.Sc Research Project
Supervisor:	Prof Vikas Sahni
Submission Due Date:	26/04/2022
Project Title:	Distributed Intrusion Detection System for Cloud Environment Using Deep Learning Machine Algorithms
Word Count:	6631 **Page Count** 24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Oyindeinbofa Pibowei
Date:	13/04/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Distributed Intrusion Detection System for Cloud Environments Using Deep Learning Machine Algorithms

Oyindeinbofa Pibowei
x20165765

Link to Video: Oyindeinbofa_Pibowei_x20165765_Research_Project_Presentation-20220421_105510-Meeting Recording.mp4

Abstract

The rapid growth of computer networks has altered the prospect of the security of networks. With the increasing connectivity between computers bring about exposure of these computers on the network to vulnerabilities. To curb the increase in attacks, intrusion detection systems were developed for the protection of computers and devices on the network. With the evolving of technology, researchers have been led to proffer evolving solutions to this challenge faced. This paper implemented a distributed intrusion detection system using deep learning machine algorithms for detection of unknown attacks on the network. Logistic Regression, Artificial Neural Network (ANN), Random Forest, and Gradient Boosting Classifiers were the implemented algorithms in this paper and was experimented on KDD99, CIC and NSL-KDD datasets to determine which condition was more suited for detection of attacks and compared to previous models carried out on the same datasets, the implemented model performed with a higher efficiency while maintaining accuracy of over 99%.

1 Introduction

1.1 Motivation and Background

Over the past decade, threats to networks have greatly increased and the effect potentially catastrophic.[1] Since the invention of smart devices and IOT, the advancement of connectivity has greatly increased also increasing the vulnerability that comes with it regarding information passed over the network. Researchers have developed previous models aimed at securing information circulated on the network and devices as big data transported over this network contains sensitive information pertaining to various organizations and enterprises. Network security has been one of the leading concerns of cloud service providers for the improvement of services offered to their clients to ensure reliability, availability and security of the services rendered to their clients.

With the development of IDS, cloud service providers (CSP) have been able to detect malicious activities carried out by cybercriminals on the network before an actual attack occurs but now face challenge of improved attacks by these intruders aimed at evading detection using sophisticated means and carry out their malicious intent hence the constant improvements of existing intrusion detection systems to detect more accurately real time attacks for the improvements of intrusion detection system. This IDS model analyzes network traffic and information to predict normal and abnormal behavior to notify the network administrator of potential attacks and proceed to prevent such attacks. Many researchers having developed models for predicting attacks face challenge when it comes to timely detection of these attacks which is very important because what need is an IDS if it detects an attack after it is carried out? This paper therefore is based on the prediction of abnormalities in the network traffic of the cloud environment.

1.2 Research Question

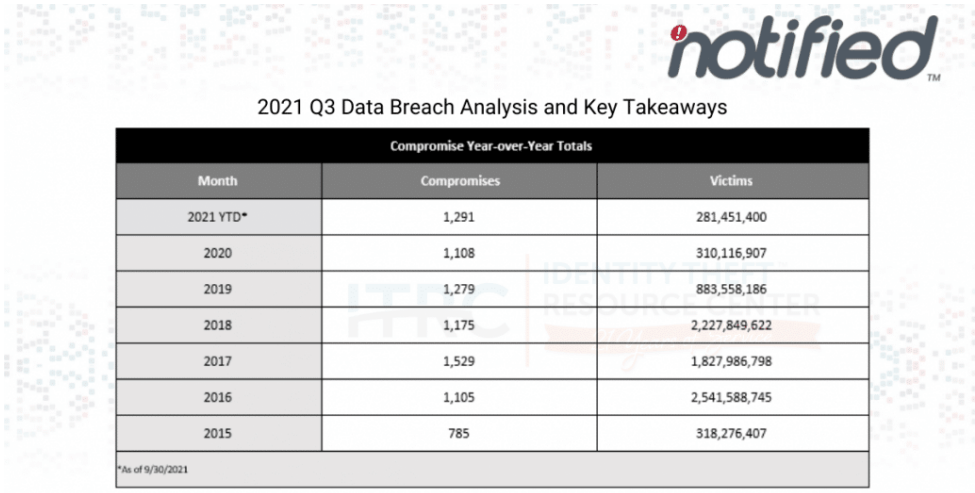
What is the improvement of an IDS model's performance built using deep learning machine algorithms for detecting intrusion in a cloud environment in terms of accuracy and precision?

Which machine learning algorithm is best suited and most accurate for a distributed intrusion detection system?

1.3 Research Objectives

Over the years from 2006-2010 the number of cyber-attacks started increasing and many businesses experienced massive decline in growth due to these attacks by cyber criminals, figure 1 shows the incident report of these attacks.[2]

And now these attack numbers have devastatingly increased in recent year causing even more damage to businesses and organization. To tackle the issue of these attacks the IDS was developed and with the development of this security measures the cyber criminals have been evolving with more sophisticated means of attack towards disrupting the service on the cloud environment and making service unavailable to the clients of cloud service providers.



Compromise Year-over-Year Totals		
Month	Compromises	Victims
2021 YTD*	1,291	281,451,400
2020	1,108	310,116,907
2019	1,279	883,558,186
2018	1,175	2,227,849,622
2017	1,529	1,827,986,798
2016	1,105	2,541,588,745
2015	785	318,276,407

*As of 9/30/2021

Figure 1: Cyber incidents in recent years

[1]

Cybercriminals have been able to evade detection due to poor performance of developed IDS and high false detection rates as such there is need for an IDS with high detection rate and low rate of false alarm to curb the increase in cyber incidents in real time. This research works towards the following objectives.

[1] https://www.idtheftcenter.org/post/identity-theft-resource-center-to-share-latest-data-breach-analysis-with-u-s-senate-commerce-committee-number-of-data-breaches-in-2021-surpasses-all-of-2020/

1. To build an IDS for detection in a cloud environment with high accuracy with low false alarm rate
2. To compare different models using deep learning machine algorithms like Random Forest, Logistic regression, Gradient Boosting Classifier and Artificial Neural Network.
3. Perform deep monitoring of the training models for an improved model in the testing environment with less time taken into consideration
4. Evaluate all models for best performance in terms of accuracy, precision, and prediction

The rest of the paper is detailed in sections as follows, Section 2 lay emphasis on related work carried out by previous researchers, Section 3 details the methodology employed for the building of the proposed model, Section 4 showcases the in depth specification of the model then Section 5 breaks down the implementation of the model using the machine algorithms employed, Section 6 evaluates and discuss the experiments results and finally Section 7 discusses the conclusion and suggests future work regarding this experiment.

2 Related Work

Many researchers carried out several experiments regarding the improvement of IDS for the detection and security of information in the cloud environment. In this section, the work done by researchers will be analysed and the experiments carried out improved upon.

2.1 Review of Related Work

Mohamed and Abdullah [3] developed an IDS with response that detected and reported abnormality of network traffic to detect attacks in the cloud environment. Their research focused on a mobile ad-hoc network (MANETs) which because of its features which allowed operation in isolated mode or in coordination with infrastructure that was fixed. This model focused on a four-stage process of operation which was detection, classification, blocking, and recovery process for the effectiveness of the security model. However, the detection process was completed by creating profiles which were the new gene, nonself-profile, and detector profile that transmits to all nodes in the domain.
This model was configured to assign each suspected attack to a profile before classification then blocking and recovery takes place. Therefore, any attack not assigned to a profile the security model would not detect, which in turn means would not advance to the classification stage of the model and would then go unchecked and that would have a drastic effect in the cloud environment and the result it produces in the testing stage of the model.

Kumar and Joshi [4] carried out the development of an IDS model using Snort, entropy, and alert ranking system. In intrusion detection, entropy is one of the detection techniques employed to analyze anomalies while Snort on the other hand is also amongst the most famous network-based IDS which makes use of an alert-based system of suspicious activities detection on the network. Alert in Snort comprises of the packet source as well as signature ID and timestamp of the destination information. The model made use of the feature extraction technique and also made use of the KDD dataset to carry out the testing of their proposed security model.

The main purpose behind the use of this dataset was that the features were calculated for entropy which helped to detect anomalies in the network traffic by the entropy-based module. Snort was set to packet sniffing mode. Snort is an alert software used for the detection of attacks in an IDS. However, Snort is susceptible to unknown attacks and also there was the flaw of identification of important alerts when bulk data was analyzed through Snort. Also, the KDD dataset has limitations as the results obtained were quite biased due to the use of all its features and not properly preparing the dataset.

Chang and Zhu [5] came up with a distributed wireless IDS model which allowed agents to communicate with other agents to best secure the cloud environment using artificial immunology algorithm which was done using clone selection, immune networks, and other immunological principles. This method was originally developed to solve the problem of data clustering. The artificial immune network algorithm (AINA) was made use of for their security model DWIDS (Distributed Wireless Intrusion Detection System) and the process involved in its operation involves the DWIDS randomly generating a certain number of network cells to the initial network before calculating the fitness of all the cells on the network to get fitness standardization.
The DWIDS generated several clone rules for each cell on the network while it kept the parent rules in the group by mutating all the clone rules which varies inversely with the parent cells' fitness. The DWIDS selected the highest fitness rules from the existing clone rules and built up a new network and generated a new fitness average for this network then calculated the affinity of all the rules in the network inclusive of the new network built up then output all the results as new rules. Although it provided basic defence of WLAN, unknown intrusions were not detected as the rules generated by this artificial immunology algorithm only applied to existing attacks and intrusions.

Another research [6] was based on two stages: the classification and identification of the incoming network traffic using RepTree algorithm. In the classification stage, incoming traffic flow was classified by its protocol if it were TCP, UDP, or other to ascertain the different features of each protocol connected to the network after which the three subsets were subjected to data pre-processing to remove noisy outliner and unrelated features. After this pre-processing, the model was able to determine if it is was an attack or if it was normal traffic on the network.
Just after the first stage of classification, a pre-trained multiclass classifier was launched whenever an attack was identified by the first classifier to determine the attack type and provided an appropriate response to the attack. One challenge this model faced in the network traffic data set was that distribution of the various protocols connected in the network was not even as some connections in some protocols were more frequent while others caused an imbalance that affects pre-processing.
Also, the use of decision tree classifiers like ID3, C4.5, CART used in this model generated large decision trees that were overfitted for the training set, limiting the performance of the classifier due to memory allocation limitations that made the detection accuracy biased.

Binbusayyis and Vaiyapuri [7] based their research on 4 datasets which included KDDCup’99, NSL-KDD, UNSW-NB15, and CICIDS2017 datasets. They established that the performance of IDS depends on the features of data to be evaluated. They started their experiment with the approach of implementing four different evaluation measures which included correlation, information, distance, and consistency to choose the features that were more crucial for intrusion detection.

After the choice of the most crucial features was done, the subset combination strategy was applied to merge the output of the four measures to arrive at the potential feature set. Also, they implemented the building of an effective IDS with a data analytic framework which comprised of 5 key phases of the data analytic lifecycle which include; data discovery, data preparation, model planning, model building, and model evaluation. These key phases of the data analytic lifecycle played a crucial role in the IDS performance to produce effective results.
Their approach to using 4 different datasets was commendable as it brought about variety in testing results to determine the effectiveness of the proposed IDS model.

Al-Yaseen, Othman and Nazri [8] researched a multi-level hybrid support vector machine (SVM) and extreme machine learning (EML) based on the modification of IDS using K-means on the existing KDDCup'99 dataset. They implemented this using 10% KDD training dataset which contained 494,021 to train the SVM and EML to reduce the training time involved when using the complete KDD training set and also to avoid the issue of memory overflow. They reduced the size of the dataset with K-means and built a new high-quality dataset which was small to ensure accuracy with the results obtained and low false alarm rate of the SVM and EML. They were able to achieve this by converting the symbolic attributes protocol, service, and flag to numeric ones then normalized the resulting data to [0,1]. After that they separated the 10% KDD training dataset into instances of 5 categories which included the normal, DoS, Probe, R2L and U2R; then they applied the modified K-means on each category and created a new training dataset from the resulting instances and followed through with using the SVM and EML with the newly created training dataset before finally testing the multi-level with the corrected KDD dataset.

Other researchers [9] argued that to solve the problem of multi-class classification problem a hybrid IDS was needed as it countered the problem faced due to the large data inflow over the network and real-time detection which according to them was almost impossible to achieve in previous models. They based their HIDS model on a Naïve Bayes feature selection which allowed their model to reduce the sample data's dimensionality. Also, this model had outliner rejection which separated the noisy input samples to avoid the issue of miscalculation. However, this outliner was prone to rejecting important data that were to be classified for detection in the training of the OSVM (Optimized Support Vector Machine). They implemented the classification through a prioritized K-nearest Neighbours classifier (PKNN)

2.2 Comparison of Related Works

Author and Publication	Methodology Employed	Dataset	Advantage	Limitation	Performance Analysis
[3]	Artificial Immune System (AIS)	3 Wireless Nodes (One Mobile and Two Stationary)	Effectively carry out monitoring , analyzing, detecting and responding to intrusion with little or no human support	Scalability and bandwidth conserving due to bulky data analyzed	For every 238 pattern there is 1 valid detector in 1306197 patterns being analzed. High detectors are cloned to

[4]	Snort, Entropy and Alert Ranking System	NSL-KDD	Combination of Signature based and Anomaly based techniques to increase performance	Snort is susceptible to unknown attacks not in the database	produce more effective agents Detection occurred in less time during experiment but due to snort susceptibility to unknown attacks, the false rate of alerts was high
[5]	Artificial Immune Algorithm (AIA)	Host AP and Network Traffic	The use of mobile agents improved the detection by communicating with each other in the testing environment.	The use of sandbox do not give room to ascertain if the model's performance in real time	This model was able to detect DDoS attacks, fake APs, also able to provide basic defence for WLAN.
[6]	RepTree Algorithm	UNSW-NB15 NSL-KDD	The application of Two stage classifier helped increase the accuracy and detection	Attack classification still not perfect as classified protocol UDP not properly classified	This model performed accurately in terms of detection speed and proffer reduced alert ranking
[7]	Ensemble Algorithm	KDD UNSW-NB15 CICIDS2017	The merging of four filter method improved the accuracy of the model	The bulky data of the dataset limited the model's performance and increased the false rate recorded	The model having implemented four filter classification performed greatly in terms of correlation and consistency
[8]	SVM ELM K-Means	KDD99	Modification of K-Means for pre-processing of training dataset	Classifiers unable to detect unknown attacks only attacks in the database	Achieved Accuracy of 95.75% and false rate of 1.87%
[9]	KNN SVM	NSL-KDD KDD99 Kyoto 2006	Implementation of Naïve Bayes Feature Selection(NBFS)	Bulky data from dataset had outliers leading to	This model had high detection rates of rare attacks like R2L and U2R

			to remove outliers	increasing in testing time	
[10]	CCAF and BPMN	Data Center	The CCAF multi-layer security can block 9,919 viruses and trojans which can be destroyed in seconds while the others can be isolated and quarantined	Due to bulky data analyzed it takes between 50-125 hours to analyze and stop a security breach	Single layered security can block a total of 7,438 viruses which makes the CCAF 20% more effective in performance
[11]	Mobile Agents in the Xen cloud environment (DOM0), (DOMU), (PV),	Xen Cloud Data	The mobile agents aided load balancing, fault tolerance to solve the issue of DoS and also it helps with network management	These mobile agents work best with virtual environments as such were limited to only VM and virtual protection	The mobile agents were able to communicate attacks and mitigate against them by authenticating each session of interaction between the CSP and the client
[12]	Self-Organizing Map (SOM), J.48 decision tree algorithm, a rule-based Decision Support System (DSS)	KDDCup'99	Anomaly detection was based on deviation from the normal model of observed data as such it was easy to detect intrusions	Misuse module misses attacks that could have been detected and had to rely on anomaly detection module for the detection of the missed attacks	The proposed model achieved good accuracy of 99.90% detection rate with a false positive rate of 1.25% and a classification rate of 98.84%.
[13]	Artificial Immune Systems (AIS), Kernel-based Virtual Machine (KVM), and Orchestra Management	NSL-KDD99	The mobile agents in this model were mobile, autonomous, they adapted and learned about new intrusions	Due to the large size of the dataset, only a small portion was used to carry out the experiment	The proposed model is a distributed intrusion detection system and was compared with a centralized

			and also communicated with other agents for the effectiveness of the security of the cloud environment.		system and the communication between agents proved that the proposed model was more effective in the detection of new attacks and mitigating against them
[14]	A backpropagation (BP) algorithm based neural network (ANN)	KDD	The fuzzy clustering dividing the training set into subsets aided the increment in performance of the IDS	The bulky nature of the dataset used posed limitations due to complexity and also the inability to handle noisy data in the dataset.	ANN achieved a high detection rate with very low-frequency attacks and stability in detection
[15]	JPCAP packet capturing, WINCAP	KDD	The effective sniffing of data both relevant and irrelevant showed the depth of detection of the proposed model as the dataset used comprised of bulk data.	Sniffers picked up irrelevant data which slowed down the detection process of the attacks	The experiment was able to detect DDoS attacks as well as capture the IP, timestamp port, and protocols (TCP, UDP, or ICMP)
[16]	Pseudocode Outlier (NOF) Neighbourhood Outlier Factor, SNORT	KDD	The NOF was effective in data clusters where attacks would normally be missed but were detected by this model	Snort had a flaw of identification of alerts when bulk data were analyzed and the KDD dataset used was one with bulk data as such a biased result was obtained	The thorough detection of the NOF increased the IDS performance by detecting all anomalies in the network traffic with an accurate rate and at a considerable time frame.

Table 1: **Comparison of related literature review's methodology and performance analysis**

The aim of an IDS is the timely detection of attacks with an effective accuracy rate which most of the previously proposed solutions lacked due to the long testing time and the bulky data being analyzed and some of the methodologies used produced biased results due to the complexity of the algorithms proposed. Also, the proposed solutions were more focused on centralized systems which disrupted the communication of agents for the detection of attacks; as a result, this paper proposes a better model which is a distributed intrusion detection system that comprises of intelligent agents that communicate and have autonomy by learning about new attacks and intrusions to best carry out detection and produce high performance with quick testing time in the cloud environment.

3 Research Methodology

Cross-Industry Standard Process for Data Mining (CRISP-DM) was adopted for the implementation of this research project to create a structural approach. The CRISP-DM in a study carried out in [17] comprised of six stages iterative framework for any data mining project Figure 2 below, illustrates the CRISP-DM methodology steps.

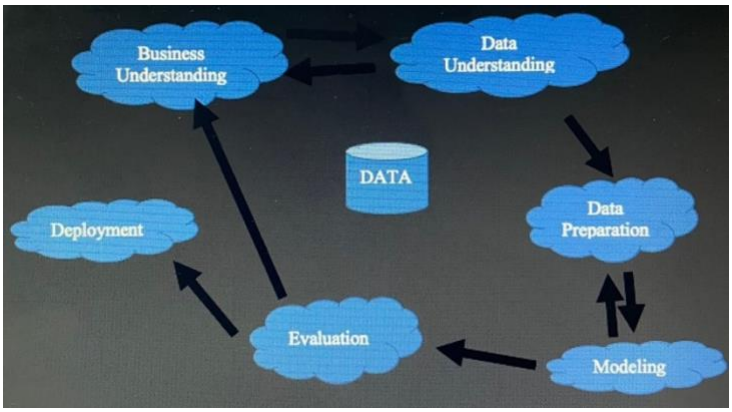


Figure 2: The Methodology steps of the CRISP-DM

3.1. Business/Problem understanding

This was the most important stage considered before beginning the development of this project. Businesses and organizations having greatly engaged in the usage of the cloud environment and its services for their activities and operations, comes the need for security to prevent financial loss and intellectual property theft from cybercriminals.
The objective of this research project was to develop a distributed IDS that would be effective in detecting attacks in the cloud environment promptly and communicating with other IDS agents to improve security.

3.2. Data Understanding

The audit of historical data which comprises known attacks and vulnerabilities was essential for the improvement of this proposed model to analyze, predict and help detect unknown instances. It was important to note the data to be used and carefully selected and also cleaned to ensure the right dataset for the implementation of this proposed model was selected.

For this project, the KDD99 dataset was selected, it contained 41 attributes, and also assigned to these attributes were labels indicating either attack or normal. This dataset is an open dataset that was made publicly available

3.3. Data Preparation

The modeling was fully dependent on the outcome of this stage, the data had to be prepared and the independent and dependent variable of the dataset had to be pre-processed and transformed to fit the requirements of the proposed model. During the preparation of this dataset, the dataset was carefully normalized and the even distribution of the class label was taken into consideration during the experiment to avoid imbalanced data in the course of preparing the dataset in order to ensure effective results were obtained. The dataset was split into 70% for training and 30% for testing and Python was the programming language used in building the models in the distributed environment; also, the SMOTE technique was used for the normalization and tackling the issue of imbalance with this data set.

3.4. Modeling

This research project modeling was implemented using machine learning algorithms for the classification of this dataset, this included logistic regression, gradient boosting classifier, random forest classifier, and artificial neural network (ANN). Using these regression and classification algorithms made it possible to properly classify each feature and data in the dataset for its relevance and removed the unwanted data and irrelevant data. In addition, noisy data was removed to reduce the time involved in the training and testing experiment carried out and achieved results that solved the limitations of previously proposed models and obtained accuracy in detection and achieved timely results.

3.5. Evaluation

The evaluation of this work was based on the accuracy, precision and recall as proposed by [18] in their proposed IDS model by means of confusion matrix which helped solve problems with classification. For this research project, two class detection problems were dealt with for which a 2x2 confusion matrix that took count of the predictive and actual values to help predict attacks and normal network traffic was considered. Figure 3 below shows the 2x2 confusion matrix:

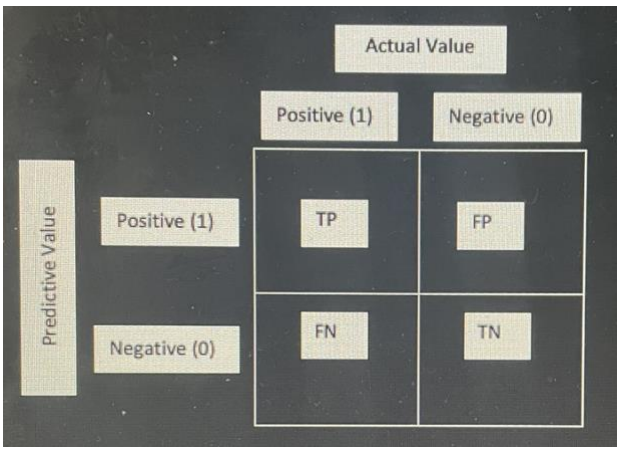


Figure 3: Confusion Matrix

For the accuracy which is the ability to obtain measurement of intrusion from the training dataset by the classifiers, it was evaluated using the metric below:

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

Also, the evaluation of the model's precision was done using the metric:

$$\text{Precision} = TP / (TP + FP)$$

Which was the measure of the correct positive prediction amongst the values of the correct predictions obtained

False Positive Rate (FPR): The proportion of normal records classified as attack records and this was evaluated below as:

$$FPR = 100 \times FP/ FP + TN$$

And lastly recall which indicates the relevant instances selected for the performance evaluation given as:

$$\text{Recall} = TP / (TP+FN)$$

All these evaluated above included pre-processing, anomaly detection, and classification time for random forest. The Receiver Operator Characteristic (ROC) and Area Under ROC (AUC) curves obtained was the same and furthermore confirms the accuracy of the evaluation performance of the proposed model. The ROC curve obtained showed the number of properly categorized positive occurrences was greater than the number of incorrectly classified negative instances. The AUC value obtained confirmed the accuracy of the proposed model.

4 Design Specification

The training and testing experiment was carried out on Intel Core i7 processor, CPU: 1.8GHz 1.99GHz, 16GB RAM, Storage: 512 GB SSD, Operating System: Windows 11, 64Bit x64 Based processor. The programming language used in this environment to process the datasets was Python 3.9.7 also Anaconda Navigator was used alongside Python and the Jupyter Notebook v6.4.8 was used for the execution. Also the use of logistic regression, gradient boosting classifier, Random Forest and Artificial Neural Network (ANN) was used to carry out the experiment regarding this research project. The specifications are given in the configuration manual.

4.1 Logistic Regression (LR)

Using Logistic regression, the error rate was reduced in the training experiment by implementing approximation to ascertain more accurate result carried out in the experiment instances. The results obtained from the training improved the performance of this proposed IDS model.

4.2 Gradient Boosting Classifier

This is a machine learning algorithm that helps with classification and regression, it was used in this research for the classification of the extracted dataset features and also to improve the predictive nature of this model due to the loss function optimization feature of the gradient boosting classifier.

4.3 Random Forest (RF)

RF was also employed for the classification of the dataset to ensure accuracy as it offers low classification error in comparison to other traditional classification algorithms. This was done by constructing decision trees during the training experiment to reduce the false alarm rate for accurate detection. RF analyses bulk data which proved effective with the datasets used in the experiment that helped to obtain high performance ratings of this model.

4.4 Artificial Neural Network (ANN)

After the training experiment was carried out, the result was converted to numeric form to carry out the testing of the ANN. This was done to ensure this model would perform effectively in the cloud environment.[18] There are two steps in the ANN testing which is the verification step and the recall step. The verification step was done by observing the ANN adapting to the result of the training experiment to learn the pattern of detection in the training phase and the recall step was done by observing the performance of the ANN in the dataset not for training to ascertain the level of learning of the neural network to effectively carry out detection and distribute to other agents in the cloud environment.

5 Implementation

In this project, various performance metrics were implemented to ascertain the performance ratings of the proposed model. During the classification stage, confusion matrix was implemented to obtain the actual detection values and then the predictive values also, after classification the correction map below shows the feature classification and removal of missing values as the dataset contained irrelevant data needed for the setting up of the experiment of this project.

5.1 Data Pre-processing

Data pre-processing steps were ensured to prepare the data for the experiment which helped in the performance of the model.

Exploratory Data Analysis (EDA): This entailed the proper understanding of the data to be used for the experiment. In this research project 3 datasets were used, KDD99 dataset, CIC dataset and the NSL-KDD dataset consisting of 494,020 records, 42 attributes, 225,745 records and 75 attributes, 163,012 records and 43 attributes respectively. During the implementation in the testing environment, the imbalance learn function was executed to balance the data in the datasets. However, the independent variables that exists in the datasets were highly correlated as shown in Figure 4 below.

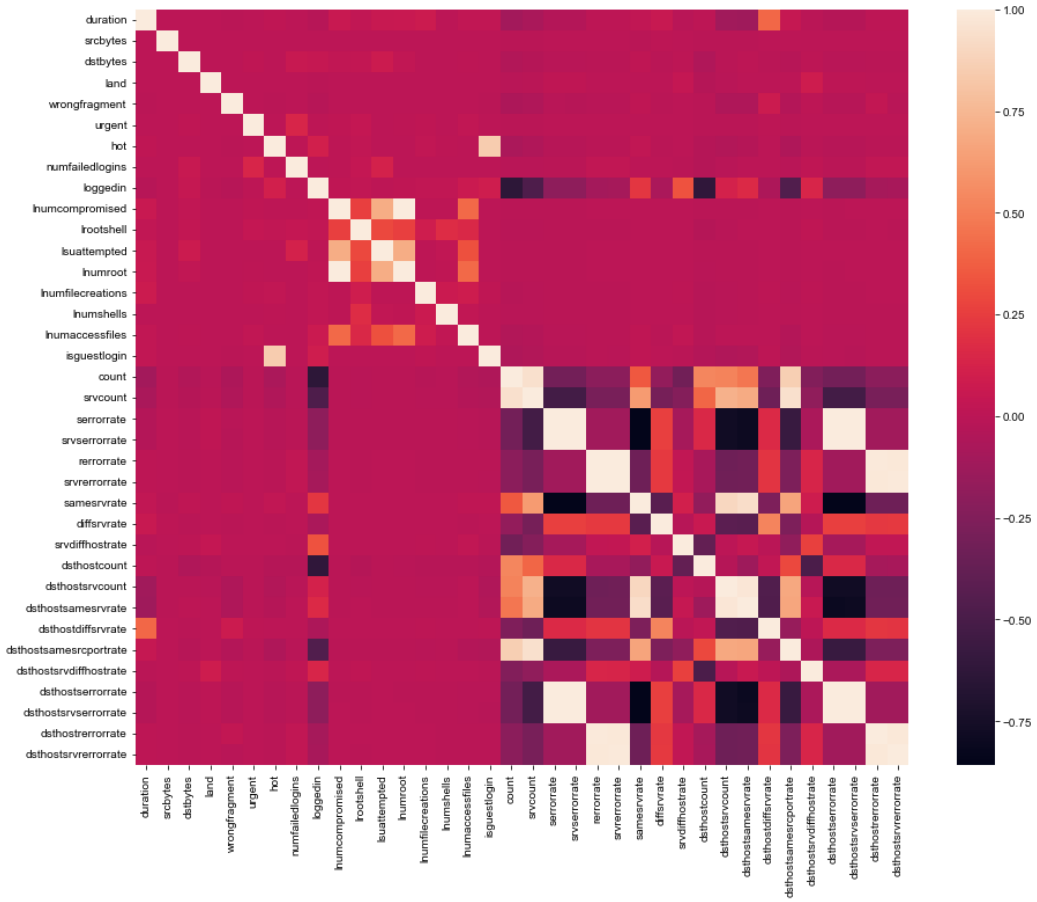


Figure 4: Correlation Heatmap

Data Transformation: The correlation was estimated without the negative correlation and the high correlation were dropped to further prepare the data to be used for the experiment.

Feature Reduction: As seen in the correlation map the dataset had many independent variables also missing values that needed to be removed to obtain a balanced dataset. This was done to ensure the issue of model over-fitting was avoided.

Feature Selection: After the reduction of the dataset features, a more balanced dataset was obtained, and this was done using Random Forest's "feature importance" function to help reduce the execution time taken to train the data while maintaining the prediction accuracy of the model.

5.2 Modeling

The proposed model was trained with various machine learning classification algorithms to obtain diversity in the performance of the model. The testing of this model was done with Random Forest, Gradient Boosting Classifier, Logistic Regression and ANN (Artificial Neural Network). The datasets were classified into attacked and non-attack, the ratio of attack to non-attack in the datasets used in the training environment is shown in Figure 4 below.

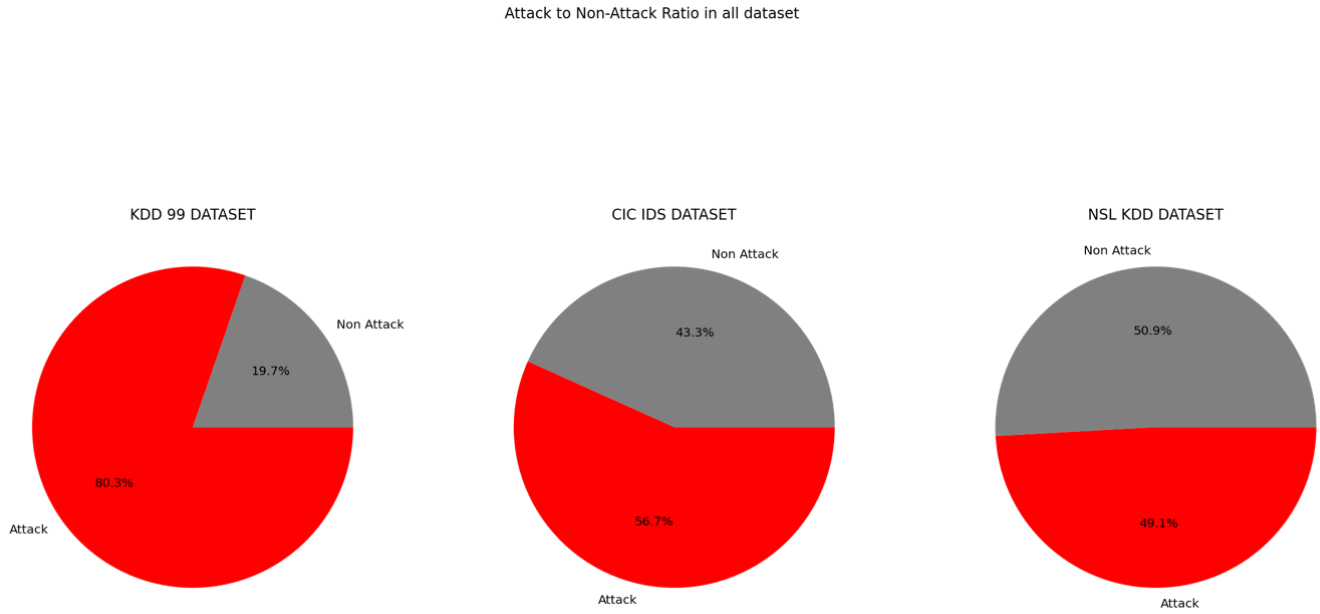


Figure 5: Attack to Non-attack Ration in all datasets

6 Evaluation

The performance of the data mining models implemented in the experiments carried out for the 3 datasets was evaluated using performance metrics namely, Accuracy, Precision, Recall and AUC. Also, confusion matrix was obtained for each classification model which is shown in the experiments evaluated below

6.1 Experiment 1: Performance of Models in KDD99 Dataset

Training Time: 8.22s
Testing Time: 0.01s

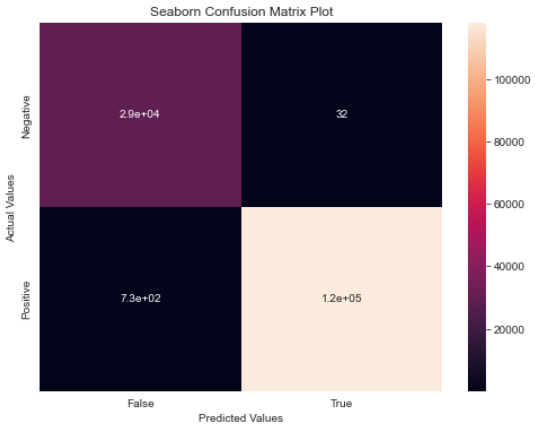


Figure 6: Confusion Matrix for LR in KDD99 dataset

Training Time: 4.22s
Testing Time: 0.42s

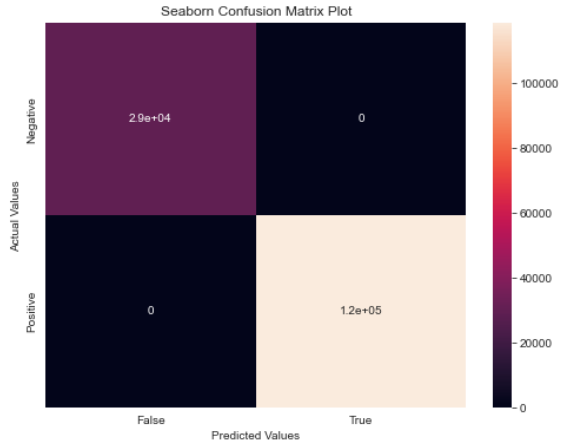


Figure 7: Confusion Matrix for RFC in KDD99 dataset

Model	Accuracy	Precision	Recall	AUC
LR	99.48%	99.97%	99.38%	99.64%
RFC	100%	100%	100%	100%
GBC	100%	100%	100%	100%
ANN	99.99%	99.98%	100%	99.97%

Table 2: Performance Metrics for KDD99 Dataset

Training Time: 28.47s
Testing Time: 0.24s

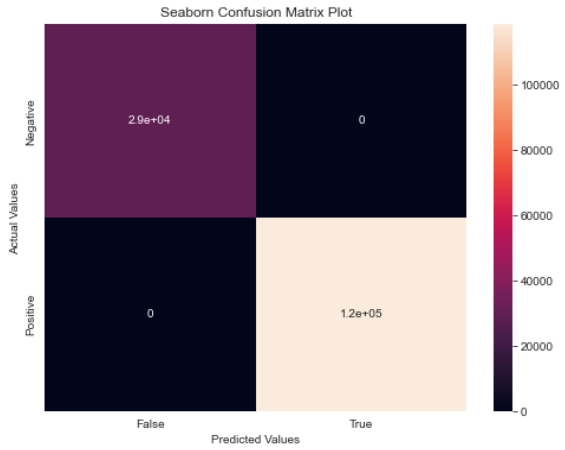


Figure 8: Confusion Matrix for GBC in KDD99 dataset

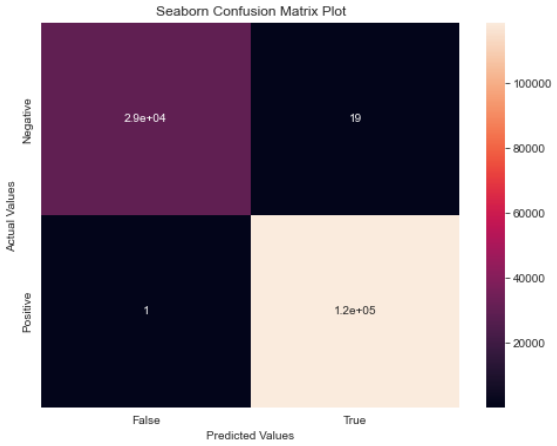


Figure 9: Confusion Matrix for ANN in KDD99 dataset

6.2 Experiment 2: Performance of Models in CIC Dataset

Training Time: 0.95s
Testing Time: 0.01s

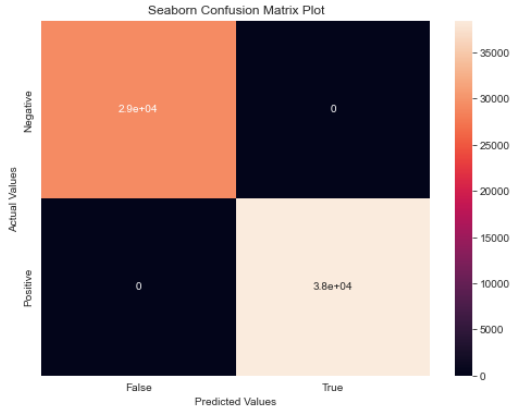


Figure 10: Confusion Matrix for LR in CIC dataset

Training Time: 3.06s
Testing Time: 0.16s

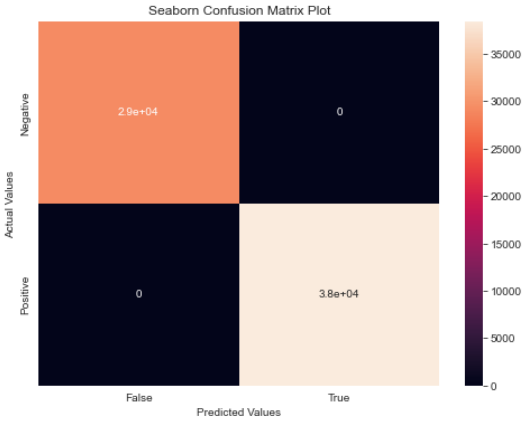


Figure 11: Confusion Matrix for RFC in CIC dataset

Model	Accuracy	Precision	Recall	AUC
LR	100%	100%	100%	100%
RFC	100%	100%	100%	100%
GBC	100%	100%	100%	100%
ANN	100%	100%	100%	100%

Table 3: Performance Metrics for CIC Dataset

Training Time: 37.21s
Testing Time: 0.14s

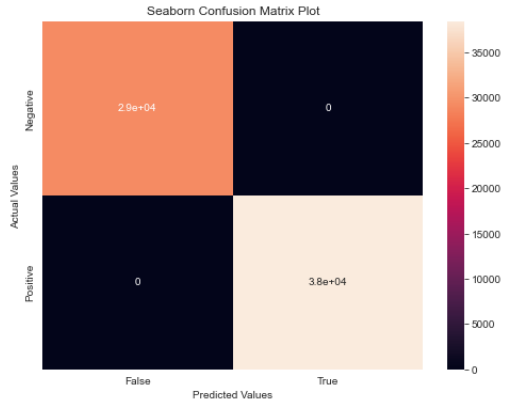


Figure 12: Confusion Matrix for GBC in CIC dataset

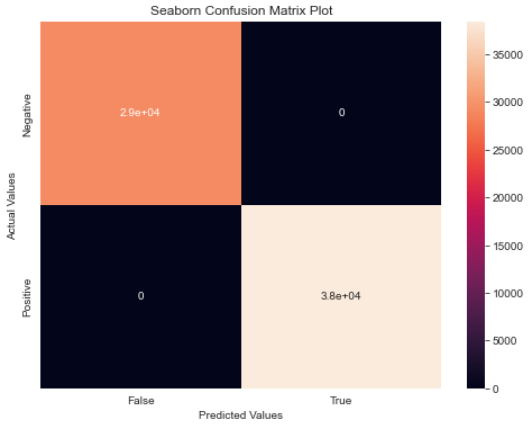


Figure 13: Confusion Matrix for ANN in CIC dataset

6.3 Experiment 3: Performance of Models in NSL-KDD Dataset

Training Time: 1.68s
Testing Time: 0.01s

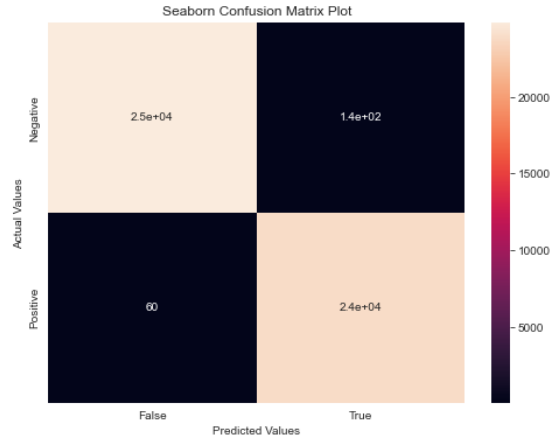


Figure 14: Confusion Matrix for LR in NSL-KDD dataset

Training Time: 2.46s
Testing Time: 0.14s

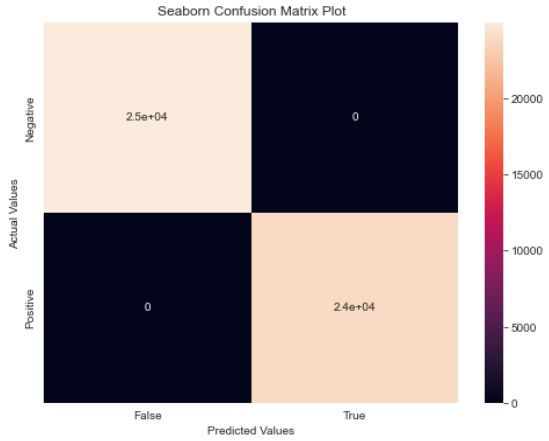


Figure 15: Confusion Matrix for RFC in NSL-KDD dataset

Model	Accuracy	Precision	Recall	AUC
LR	99.6%	99.43%	99.75%	99.60%
RFC	100%	100%	100%	100%
GBC	100%	100%	100%	100%
ANN	100%	100%	100%	100%

Table 4: Performance Metrics for NSL-KDD Dataset

Training Time: 11.37s
Testing Time: 0.08s

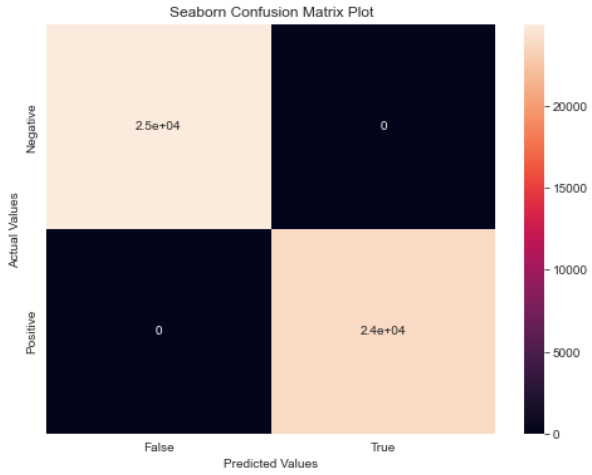


Figure 16: Confusion Matrix for GBC in NSL-KDD dataset

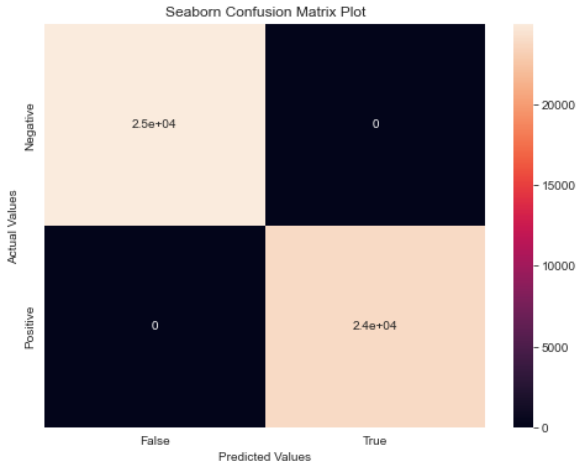


Figure 17: Confusion Matrix for ANN in NSL-KDD dataset

6.4 Discussion

All the models of classifications performed with high accuracy with all three dataset and produced a low error rate during the experiment. However, the Logistic regression model in the NSL-KDD and the KDD99 datasets performed less with accuracy of 99.6% and 99.48% respectively compared to the CIC dataset with accuracy of 100% with training time of 0.95s and testing time of 0.01s. The training of the Artificial Neural Network to detect attack patterns showed the learning accuracy of the model and the ability for it to act autonomously in the cloud environment. When compared to previous research carried out in [12] involving naïve bayes and decision tree models on the KDD99 dataset, the models involved in this experiment were more accurate in terms of classification and accuracy with the decision tree model having accuracy of 99% and the Gradient boosting classifier at 100% detection accuracy also with less training and testing time 28.47s and 0.24s respectively.
Regardless of the obtained accuracy results in the experiments carried out on all three datasets, there is still room for communication improvement with the agents in the proposed model to help with detection of unknown attacks to provide more security in the cloud environment.

7 Conclusion and Future Work

Which machine learning algorithm is best suited and most accurate for a distributed intrusion detection system? The experiment carried out in this research project showed that the machine learning algorithms employed for the experiment performed highly in the dataset experimented on, but ANN performed best in terms of classification rate and detection rate as well as the reactivity and autonomous behavior of the classification models employed as these characteristics define a well-trained model that would perform effectively in a distributed environment. The challenged faced in this experiment was the transformation of the data from categorized data to numerical data which was suited for the training of the ANN. This model can be improved by further removing features and highly correlated data to reduce the testing time as the timing in very essential for attack detection in real-time in the network environment.

Acknowledgement

I would like to give my sincere thanks to my research supervisor, Prof Vikas Sahni for his guidance and knowledgeable contribution to the success of this research project also I would like to specially thank my family and my fiancée for their encouragement and finally I would like to appreciate the faculty of computing National College of Ireland for the opportunity to learn and develop my cybersecurity knowledge.

References

[1] F. Sabahi and A. Movaghar, "Intrusion Detection: A Survey," 2008 Third International Conference on Systems and Networks Communications, 2008, pp. 23-26, doi: 10.1109/ICSNC.2008.44.

[2] A. S. Ashoor, and S. Gore, "Importance of intrusion detection system (IDS)" In *2011 International Journal of Scientific and Engineering Research, 2*(1), 1-4.

[3] Y. A. Mohamed and A. B. Abdullah, "Implementation of IDS with the response for securing MANETs," in *2010 Int. Symp. Inf. Technol., Kuala Lumpar, Malaysia, June 15-17, 2010*, pp. 660-665. doi: 10.1109/ITSIM.2010.5561608.

[4] S. Kumar and R. C. Joshi, "Design and implementation of IDS using Snort, Entropy and Alert ranking system," in *2011 Int. Conf. Signal Process., Commun., Comput., Netw. Technol., Thuckalay, India, July 21-22, 2011,* pp. 264-268, doi: 10.1109/ICSCCN.2011.6024556.

[5] Z. Chang and Y. L. Zhu, "The design of wireless intrusion detection system based on an immune algorithm," in *2011 Int. Conf. Mach. Learn. Cybern., Guilin, China, July 10-13, 2011,* pp. 561-565. doi: 10.1109/ICMLC.2011.6016834.

[6] M. Belouch, S. El Hadaj, and M. Idhammad, "A two-stage classifier approach using RepTree algorithm for network intrusion detection," *Int. J. Adv. Comput. Sci. Appl.* (*IJACSA*), vol. 8, no. 6, pp. 389-394, 2017. doi: 10.14569/IJACSA.2017.080651.

[7] A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach," *IEEE Access*, vol. 7, pp. 106495-106513, July 2019. doi: 10.1109/ACCESS.2019.2929487.

[8] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system", *Expert Syst. Appl.*, vol. 67, pp. 296-303, Jan. 2017. doi: 10.1016/j.eswa.2016.09.041.

[9] A. I. Saleh, F. M. Talaat, and L. M. Labib, "A hybrid intrusion detection system (HIDS) based on prioritized k-nearest neighbors and optimized SVM classifiers," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 403–443, Mar. 2019. doi: 10.1007/s10462-017-9567-1.

[10] M. Ramachandran and V. Chang, "Towards performance evaluation of cloud service providers for cloud data security," *Int. J. Inf. Manage.*, vol. 36, no. 4, pp. 618–625, Aug. 2016. doi: 10.1016/j.ijinfomgt.2016.03.005.

[11] P. Singh Hada, R. Singh, and M. Manmohan Meghwal, "Security agents: A mobile agent-based trust model for cloud computing," *Int. J. Comput. Appl.*, vol. 36, no. 12, pp. 12-15, Dec. 2011.

[12] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, Nov. 2005. doi: 10.1016/j.eswa.2005.05.002.

[13] N. Afzali Seresht and R. Azmi, "MAIS-IDS: A distributed intrusion detection system

using multi-agent AIS approach," *Eng. Appl. Artif. Intell.*, vol. 35, pp. 286–298, Oct. 2014. doi: 10.1016/j.engappai.2014.06.022.

[14] H. Wang, H. Zhou, and C. Wang, "Virtual machine-based Intrusion Detection System framework in cloud computing environment," *J. Comput.*, vol. 7, no. 10, pp. 2397-2403, Oct. 2012. doi: 10.4304/jcp.7.10.2397-2403.

[15] P. Shinde and T. J. Parvat, "DDoS attack analyzer: Using JPCAP and WinCap," *Procedia Comput. Sci.*, vol. 79, pp. 781–784, 2016. doi: 10.1016/j.procs.2016.03.103.

[16] J. Jabez and B. Muthukumar, "Intrusion Detection System (IDS): Anomaly detection using outlier detection approach," *Procedia Comput. Sci.*, vol. 48, pp. 338-346, 2015. doi: 10.1016/j.procs.2015.04.191.

[17] B. Singh, "CRISP Model: A structured approach for presentation of research," *CSI Commun.*, vol. 42, no. 7, pp. 11-17. doi: 10.13140/RG.2.2.10159.79527.

[18] P. Chandra, U. K. Lilhore, and N. Agrawal, "Network intrusion detection system based on modified random forest classifiers for KDD Cup-99 And NSL-KDD dataset," *Int. Research J. Eng. Technol. (IRJET)*, vol. 4, no. 8, pp. 786–791.

[19] S.Iqbal, M.L. M.Kiah, B. Dhaghighi, M.Hussain, S.Khan, M.K.Khan, K.-K.R. Choo, On cloud security attacks: A taxonomy and intrusion detection and prevention as a service, Journal of Network and Computer Applications 74 (2016) 98–120.

[20] Wikipedia, 2016 dyn cyber attack [Online; accessed 10-November-2017)].

[21] Z. Li, W. Sun, L. Wang, A neural network-based distributed intrusion detection system on the cloud platform, in Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on, Vol. 1, IEEE, 2012, pp. 75–79.

[22] R. C. Cavalcante, I. I. Bittencourt, A. P. Da Silva, M. Silva, E. Costa, and R. Santos,""A survey of security in multi-agent systems"" Expert Systems with Applications, vol. 39, no. 5. pp. 4835–4846, 2012.

[23] Z. A. Baig,""Multi-agent systems for protecting critical infrastructures: A survey""Journal of Network and Computer Applications, vol. 35, no. 3. pp. 1151–1161, 2012.

[24] M. Ring, S. Wunderlich, D. Grdl, D. Landes, A. Hotho, Flow-based benchmark data sets for intrusion detection, in Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS), ACPI, 2017, pp. 361–369.

[25] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa,""JADE: A software framework for developing multi-agent applications. Lessons learned"" Inf. Softw. Technol., vol. 50, no. 1–2, pp. 10–21, 2008.

[26] C. J. Su,""Mobile multi-agent based, distributed information platform (MADIP) for wide-area e-health monitoring"" Comput. Ind., vol. 59, no. 1, pp. 55–68, 2008.

[27] G. Fortino, A. Garro, and W. Russo,""Achieving Mobile Agent Systems interoperability through software layering"" Inf. Softw. Technol., vol. 50, no. 4, pp. 322–341, 2008.