University of Ibadan
Journal of Science and Logics in ICT Research

Development of a Machine Learning Model to Enhance Incremental Approaches for Anomaly Detection in a Network

[1]OGUNTUNDE, T., [2]OLA, M. O.

Computer Science Department, University of Ibadan, Nigeria.
[1]Toyin.oguntunde@gmail.com, [2]olamargaret20@gmail.com

Abstract

The widespread availability of advanced networking technologies has led to a high rate of threat from spammers, intruders or attackers, and criminals. Over years, attempts have been made by System administrators to prevent network attacks using available signature-based Intrusion Detection Systems (IDSs). A special type of IDSs, called Anomaly Detection Systems are capable of detecting both known and unknown attacks and able to work in online mode but with challenges of a high rate of false alarm. Therefore, this research work aimed at developing a model to enhance an incremental approach for anomaly detection in a network using a One-class Support Vector Machine to improve the classification for novel attacks and as well reduce the false alarm rate of the system. A model for detection and classification of network anomaly was developed using a One-class Support vector machine (OCSVM) and Incremental approach to reasonably reduce the false alarm rate by building a model suitable for a rea-time network with the use of KDD'99 datasets to create a fast, scalable and adaptive anomaly detection. The dataset has 494,021 observations which contain 24 training attack types, with an additional 14 types in the test data only. This research work provides a scope that is possible to identify network anomaly using default Sklearn's OC-SVM parameter values and varying values for the "gamma" Parameter. In the One-class Support Vector Machine (OCSVM) technique, the network anomaly was predicted accurately at a 95 percent accuracy rate.

.

Keywords: *Computer network attacks, Incremental approach, Intrusion detection systems, One-Class Support Vector Machine, hyperparameters*

1. INTRODUCTION

The introduction of technologies such as 3G and pervasive computing has created flexibility in the interconnection of network devices ranging from mobile phones to distributed computers. Recent growth in communication organizations has led to the interconnection of distant corners of the world using advanced network technology, intruders or attacker's activities have also increased on networking infrastructure commensurately [2]. Network security is sets of conditions, constraints, and settings that allow a network administrator to prevent and monitor unauthorized access, modification in the system, misuse, or denial of a computer network and network-accessible resources. Network security involves policies that are used to manage authorization of access to data in a network, which is controlled by the network admin. It has become more important to personal computer users, and organizations to establish a form of access control. According to Joshi et. al., [5], A firewall enforces user's access policies such as what services are allowed to be accessed for network users to prevent unauthorized access to the system, this component may fail to check potentially harmful content such as computer worms or Trojans being transmitted over the network. Anti-virus software or an intrusion detection system (IDS) helps detect malware. Today anomaly detection software is also employed to monitor the network like wire shark traffic and may be logged for audit purposes and for later on high-level analysis in the system.

An intrusion detection system using an incremental approach has been widely employed

OGUNTUNDE T., OLA M. O., (2021). Development of a Machine Learning Model to Enhance Incremental Approaches for Anomaly Detection in a Network. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*, Vol. 6 No. 2, pp. 68-78

in an anomaly-based intrusion detection system. However, there is a high rate of false-alarm, they are non-scalable, and are not fit for deployment in high-speed networks [6]. Therefore, there is a need to develop a machine learning model to reduce the high rate of false-alarm, be more scalable and fit for high-speed networks to enhance the incremental approach using one-class support vector machine which can be used for network intrusion detection and to increase the overall performance of the system.

2. Literature Review

This section provides the theoretical background which gives proper insight into the review of literature relevant to this research work. The essence of this review is to provide proper background knowledge by giving a more detailed explanation of the important areas of network security which include Intrusion detection systems and Network anomaly detections, using supervised or unsupervised machine learning models for network anomaly detection which has their backgrounds in Artificial Intelligence and Machine Learning, are explained in this section.

The widespread availability of advanced networking technologies has resulted in a high rate of threat from spammers, intruders or attackers, and criminals on enterprise networks [2]. However, there is a significant lack of security methods that can be easily implemented. There exists a "communication gap" between the developers of security technology and developers of networks. Network design is a developed process that depends on the Open Systems Interface (OSI) model. The OSI model has several advantages when designing network security. It offers modularity, ease of use, flexibility, and standardization of protocols. The protocols of different layers can be easily combined to create stacks that allow modular development. In contrast, secure network design is not a well-developed process [4].

There isn't a methodology to manage the complexity of security requirements. When considering network security, it should be emphasized that the complete network is secure. It does not only concern the security in the computers at each end of the communication chain. When transferring from one node to another node data the communication channel should not be vulnerable to attack. A hacker will target the communication channel, get the data, and decrypt it and re-insert a duplicate message.

Nitin [10], stated that Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of possible incidents, which are violations or imminent threats of violation of computer security policies, acceptable use policies, or standard security practices. An intrusion refers to any unauthorized access or malicious utilization of information resources. An intruder or an attacker is a real-world entity that tries to find a means to gain unauthorized access to information, causes harm, or engages in other malicious activities.

Several types of IDS technologies exist due to the variance of network configurations. Each type has advantages and disadvantages in detection, configuration, and cost. Mainly, there are three important distinct families of IDS: Are network-based Intrusion Detection System, host-based Intrusion Detection System, and Wireless system.

Intrusion prevention is an amalgam of security technologies. Its goal is to anticipate and stop the attacks. The intrusion prevention is applied by some recent IDS. Instead of analyzing the traffic logs, which lies in discovering the attacks after they took place, intrusion prevention tries to warn against such attacks. While the systems of intrusion detection try to give the alert, the intrusion prevention systems block the traffic rated dangerous [3].

For Mudzingwa [9], there are three major techniques of detection used for Intrusion based detection systems which include: Signature-based detection, Anomaly-based detection, and Stateful Protocol Inspection. An IDS can use signature-based detection, relying on known traffic data to analyze potentially unwanted traffic. This type of detection is very fast and easy to configure. However, an attacker can slightly modify an attack to render it undetectable by a signature-based IDS. Still, signature-based detection, although limited in its detection capability, can be very accurate. Moreover, an IDS that looks at network traffic and detects incorrect data, not valid, or generally abnormal is called anomaly-based detection. This method is useful for detecting unwanted traffic that is not specifically known. For instance, anomaly-based

IDS will detect that an Internet protocol (IP) packet is malformed. It does not detect that it is malformed in a specific way but indicates that it is anomalous. In contrast to this, Stateful protocol inspection is similar to anomaly-based detection, but it can also analyze traffic at the network and transport layer and render-specific traffic at the application layer, which anomaly-based detection cannot do.

An IDS that looks at network traffic and detects incorrect data, not valid, or generally abnormal is called anomaly-based detection. This method is useful for detecting unwanted traffic that is not specifically known. For instance, anomaly-based IDS will detect that an Internet protocol (IP) packet is malformed. It does not detect that it is malformed in a specific way but indicates that it is anomalous. An anomaly-based intrusion detection system is a technique for detecting both network and computer intrusions and misuse by monitoring system activity and classifying it as either normal or anomalous [11].

An IDS can use signature-based detection, relying on known traffic data to analyze potentially unwanted traffic. This type of detection is very fast and easy to configure. However, an attacker can slightly modify an attack to render it undetectable by a signature-based IDS. Still, signature-based detection, although limited in its detection capability but can be very accurate. Signature-based IDS refers to the detection of attacks by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware. The terminology is generated by anti-virus software, which refers to these detected patterns as signatures

Anomaly detection refers to the automatic identification of unusual patterns from a large amount of data that do not conform to expected behavior. These non-conforming patterns are generally referred to in different application fields as anomalies, aberrations, discordant observations, exceptions, novelty, outliers, peculiarities or contaminants, surprises, strangeness [7]. There have been several application fields from intrusion detection, e.g. identifying strange patterns in network traffic that could signal a hack to the system, health monitoring (spotting a malignant tumor in an MRI image scan) and from fraud detection in credit card transactions to fault detection in operating environments [8]. In practice, it is very difficult to precisely detect anomalies in network traffic or normal data. So, an anomaly is an interesting pattern due to the effect of traffic or normal data, while noise consists of non-interesting patterns that hinder traffic data analysis

3. Methodology

In an attempt to find anomalous behavior in network packets using machine learning tools by employing a model that can classify both the known and novel attack usually regarded as a zero-day attack, an incremental anomaly detection model used in this research work allow the model to detect known and unknown attacks and thereafter reducing the false alarm rate.
The methodology adopted in this work is discussed in this section. It is divided into three major sections which include the discussion of the methodology used for the design, the detailed description of the system components, and the evaluation metrics used for the design.

3.1 *System Architecture of Anomaly Detection*

The system architecture of the anomaly detection system used for the research work is shown in Figure 1. The architecture is divided into three modules which include the Packet Capture Module, the Classification Module, and the response Module.

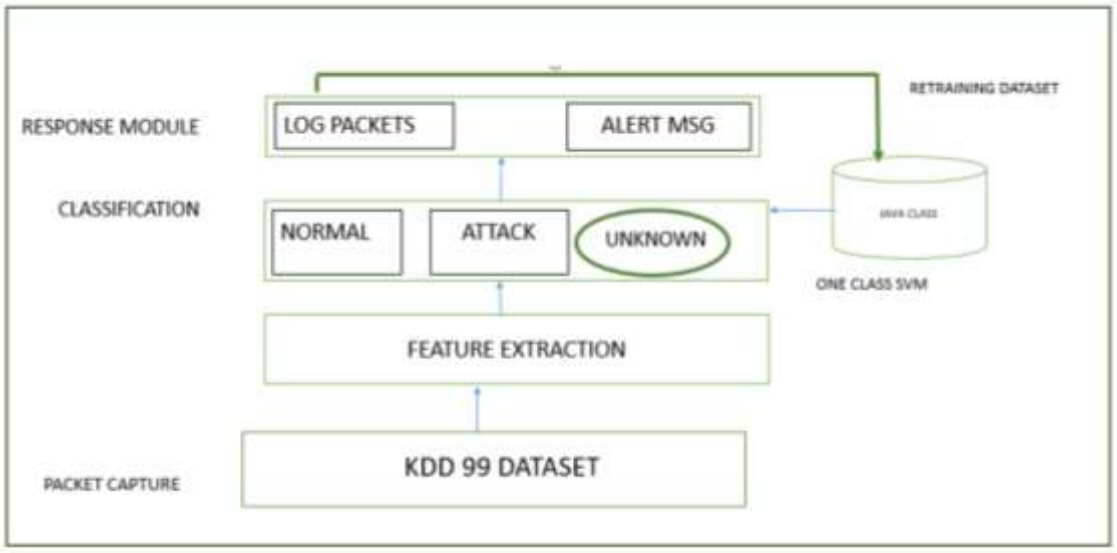

Figure 1: Architecture of Incremental Approach Intrusion Detection System (IDS)

3.2 The Packet Capture Module

The packets used for analyzing the network for infiltration and anomalous situation are presented in this module. Based on the evaluation of the anomaly-based intrusion detection system in Lincoln Lab, DARPA training and testing dataset were generated. The dataset consists of attacks and background traffic. The dataset is known as the KDD99 data set. For an experimental reason, the model developed in this research work makes use of this dataset for developing an enhanced incremental approach for anomaly detection in network traffic.

3.3 Classification Module

This module is shown in Figure1 consist of the classification algorithms used for classifying the network packets into attacks and non-attacks categories with others that are not properly labeled as unknown. Support Vector Machines (SVM) is employed because they are a special type of Machine learning techniques that are computationally inexpensive and provide a sparse solution and since the focus of this research is to enhance the existing anomaly detection system by creating a model that is suitable for deployment in a real-time network environment and with a low false alarm rate (FAR). The OCSVM algorithm maps input data into a high dimensional feature space (via a kernel) and iteratively finds the maximal margin hyperplane which best separates the training data from the origin.

3.4 Response Module

An important aspect of any anomaly detection technique is the way it reports anomalies. In this section, the classification model presents an output for the testing dataset by classifying the packet into attacks, and non-attacks, as shown in Figure1, the packets with anomalous properties contrary to the features presented in the testing dataset used for the model, will the flagged as attacks. However, the algorithm makes it possible to detect and classify novel packets which are not part of the training set but are present in the testing dataset.

The methodology employed for this research work is divided into three major parts namely:

1. Data collection and preprocessing
2. Model design and training
3. Anomaly detection and evaluation

3.5 Data Collection and Preprocessing

This section of the research work explains the methodology involved in the collection of the dataset used for the research work and how the dataset was processed to make it ready for use in our analysis. The KDD'99 datasets were used for the design to create a fast, scalable and adaptive anomaly detection. It has 494,021 observations (packets).

3.6 KDD99 Training and Testing Dataset

It is important to note that the test data is not from the same probability distribution as the training data, and it includes specific attack types not in the training data which makes the task more

realistic. Some intrusion experts believe that most novel attacks are variants of known attacks and the signature of known attacks can be sufficient to catch novel variants. The datasets contain a total number of 24 training attack types, with an additional 14 types in the test data only.

3.7 Attack Types of KDD'99 Dataset

The name and detail description of the training attack types are listed in KDD'99 features are classified into 23 predicted labels as shown in Table1 with a total of 494,021 observations, a training set of 330,995 observations, and a testing Set of 163,026 observations. Several types of attacks observed in the dataset are presented in Table 1.

3.8 Features of the KDD'99 dataset

KDD'99 Dataset consists of files generated during the data collection which will be organized in a standard format that contains 41 features for each registered connection. A connection here refers to a sequence of TCP packets with a well-defined time duration and transmitted over a well-defined protocol between a source machine and a destination machine. Each connection is labeled as either normal or under a specific sort of attack. Each connection register is about 100 bytes long. Some of the selected 41 features and their descriptions are shown in Table 2.

Table 1: Classification of the Packets into 23 Predicted Labels

SN	Attack Label	No of Packets
1	smurf.	280790
2	Neptune.	107201
3	normal.	97278
4	back.	2203
5	satan.	1589
6	ipsweep.	1247
7	port sweep.	1040
8	warezclient.	1020
9	teardrop.	979
10	pod.	264
11	Nmap.	231
12	guess_passwd.	53
13	buffer_overflow.	30
14	land.	21
15	warezmaster.	20
16	IMAP.	12
17	rootkit.	10
18	load-module.	9
19	ftp_write.	8
20	multihop.	7
21	phf.	4
22	Perl.	3
23	spy.	2

Table 2: Selected Features of KDD '99 Dataset (Source: Balakrishnan, 2014)

ID	Feature Name	Description	Type
10	count	number of connections to the same host as the current connection in the past two seconds	Continuous
Note: The following features refer to these same-host connections.			
11	serror_rate	% of connections that have "SYN" errors	Continuous
12	rerror_rate	% of connections that have "REJ" errors	Continuous
13	same_srv_rate	% of connections to the same service	Continuous
14	diff_srv_rate	% of connections to different services	Continuous
15	srv_count	number of connections to the same service as the current connection in the past two seconds	Continuous
Note: The following features refer to these same-service connections.			
16	srv_serror_rate	% of connections that have "SYN" errors	Continuous
17	srv_rerror_rate	% of connections that have "REJ" errors	Continuous
18	srv_diff_host_rate	% of connections to different hosts	Continuous

4. Results and Discussion

The experiment was modeled using KDD "99 datasets with a total of 494,021 observations which are used as samples in the testing and the training stage of the work. The dataset has undergone the stages in data preprocessing. This section discusses extensively the result generated from each stage.

4.1 Data Preprocessing

The first important deficiency in the KDD data set is the huge number of redundant records. Analyzing KDD training and testing sets, duplicate records were extracted from the training set and the testing set. Since a huge amount of redundant records in the training set can cause learning algorithms to be biased towards the more frequent records and thus prevent it from learning infrequent records which are usually more harmful to networks such as User to Root Attacks (U2R).

The existence of these repeated records in the testing set, on the other hand, will cause the evaluation results to be biased by the methods which have better detection rates on the frequent records. The first processing to be carried out is to relabel the dataset into anomalous and normal packets. In addition, to analyze the difficulty level of the records in the KDD data set, an active learning process was used to label the records of the entire KDD training and testing sets, which gives 21 predicted labels for each of the records. The reason we got these statistics on both the KDD train and test sets is that in many research works, random parts of the KDD train set are used as test sets. This research work shall provide a solution to solve the two mentioned issues, resulting in new train and test sets that consist of selected records of the complete KDD data set.

The number of records in the train and test sets is reasonable. This advantage makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. The use of active learning helped us to pre-process the dataset to a quality that is usable by the one-class support vector machine. This is very important because we need a dataset that is clean and non-redundant both in the train and test sets to allow us to generate an accurate detection result. All the 21 predicted labels were grouped into anomalous while all the non-anomalous were simply labeled as normal. The grouping is an important and necessary step for the One-Class SVM Algorithm that we have chosen for this work. Just as the name implies "One Class" the model only trains on one class such that during classification any out-of-sample observation that the model cannot recognize shall be classified as an anomaly.

Other pre-processing stages were carried out such as encoding of categorical features and normalizing of data type. The dataset used for this model has a lot of string data types as the categorical features, there is a need to carry out feature encoding (One-hot encoding) using pandas get dummy() function. Also, the original

columns are dropped or replaced with the encoded ones. To perform this, the first column of each encoded categorical feature column was dropped (to prevent multicollinearity, duplication of derived features which could make some feature dominate others thereby creating a bias in our model during training.

It is important here to check for missing values in the features contained in the data set and ensured that all records are completely perfect. Value count is used from the model library to display and count all unique values for each feature, to display any omitted categorical feature during the feature encoding. After this stage is completed then, the training set was used to train the model.

4.2 Model Design and Training

After selecting and cleaning the dataset, the next phase in the methodology is the design of the model. The machine learning algorithm used is the One-Class Support Vector Machine. It is an algorithm that is used on unsupervised data. The model is discussed in the section below. Support Vector Machines are supervised learning models with associated learning algorithms that analyse data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Now that we have established the computation of optimal hyperplane using suitable hyperparameters of the support vector machine as a model for classification, the next stage is to illustrate the decision to use One-Class Support Vector Machine (OCSVM) as the algorithm for the classification problem. One-Class SVMs have been devised for cases in which one class only is known and the problem is to detect anything outside this class. Since our KDD '99 dataset has been labeled into two distinct classes of normal and anomalous packets. The OCSVM was trained with normal packets such that every other observation outside the learned features was classified as anomalous packets. However, the model was designed to learn each observable feature in the dataset such that a packet with similar observation can as well be classified as a normal packet thereby reducing the rate of false prediction. The new packets introduced in the testing set are known as novelty detection which could be normal packets or attacks depending on the features contained in the packet and it refers to automatic identification of unforeseen or abnormal phenomena i.e. can easily identify outliers embedded in a large amount of normal data.

This model learned a decision boundary that achieves maximum separation between the samples of the known class and the origin. Only a small fraction of data points is allowed to lie on the other side of the decision boundary: those data points are considered as outliers.

The designed model needs to be trained on how to identify network anomalies. This was done by dividing the dataset into two parts namely:

1. Training set,
2. Test set.

*4.3 Training Set***:** This is a part of the dataset that is used for training the designed model. This part comprises 2/3 (i.e. 67%) for the training set of the total dataset.

*4.4 Test Set***:** This is a part of the dataset that is used to test the trained model. It is not used in the training and validation process. It comprises a further 1/3 (i.e. 33%) of the dataset and the outcome is used in the evaluation of the model to determine how well the model has performed.
It is after the model has been trained and validated that we now use it for the classification problem. The whole process is implemented on the test data that the model has not seen before to remove bias and overfitting of the designed model.

4.5 Evaluation of the result

To justify the efficiency of the proposed solution, the work was benchmarked with the most recent related work based on:

i. Success rate: the methodology used for this research was compared with other algorithms to show the resultant effectiveness of each algorithm when compared with the result gotten.
ii. Ease of implementation: the methodology used for this model was compared with other

algorithms based on ease of implementation to determine which is easier and more efficient to implement.

The model developed was also evaluated using the Classification Report and Confusion Matrix method under different performance metrics such as Accuracy, Area Under Curve (AUC), (k), Precession, and F -Measure. The results obtained for each model were shown using Classification Report and Confusion Matrix (plotted). The results obtained for each model are shown in Figure 2.

The report in Figure 2 shows that the model using the default parameters with all the 41 features in the dataset performs with an accuracy of 90% after the False Alarm Rate (FAR) was calculated.

The confusion matrix shown in figure 2 indicate that 130,423 of the test set packets were classified as True positive, 16,222 as True Negative, 15,945 packets as False Positive while 437 packets are predicted as False Negative.

The classification model designed for this research work allows the retraining of the entire model with the novel or newly classified packets without having to retrain the model with the entire training set. The incremental modes of retraining the model make it efficient in detecting and properly classifying an unknown attack that is not captured in the training set used to train the model. The general classification report of the model showing the precision, recall, f1-measure, and Area under the curve is shown in Figure 3.

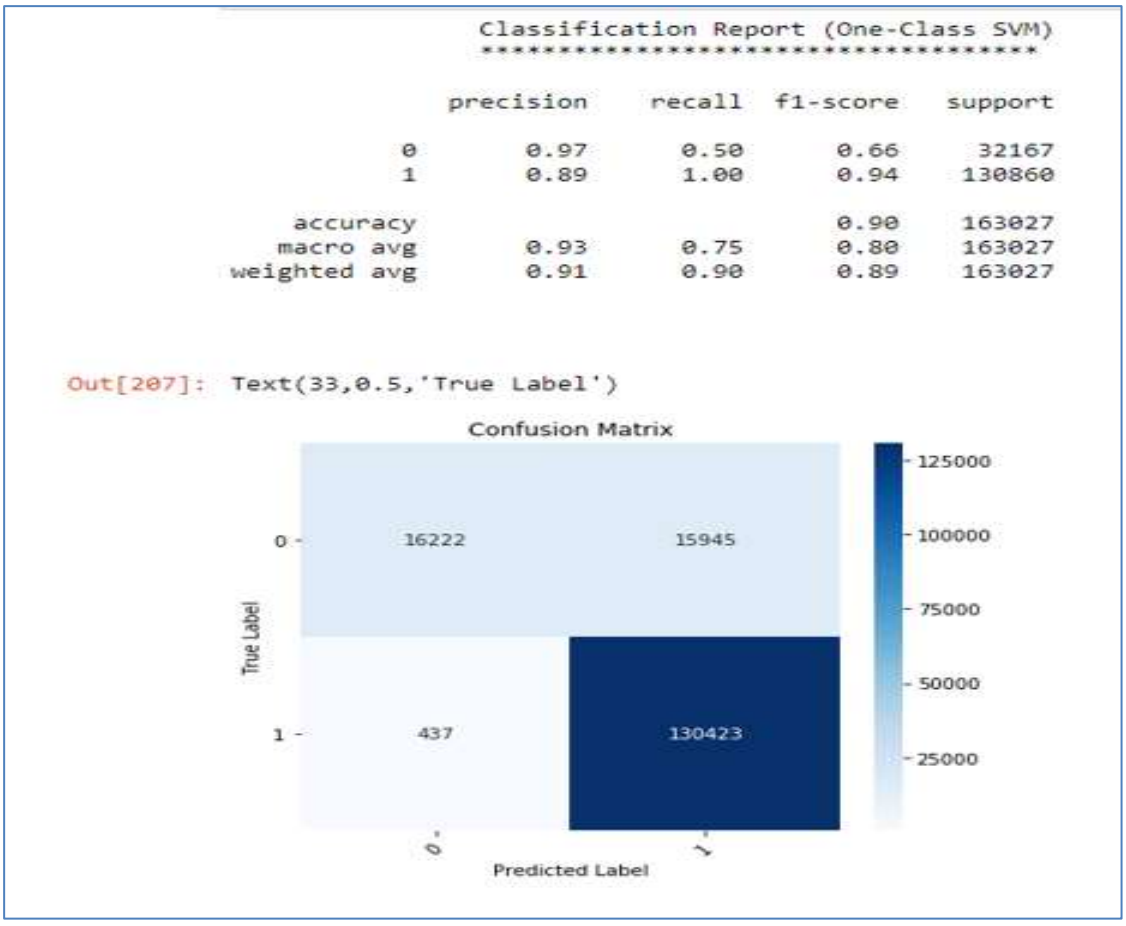

Figure 2: Classification of OCSVM model

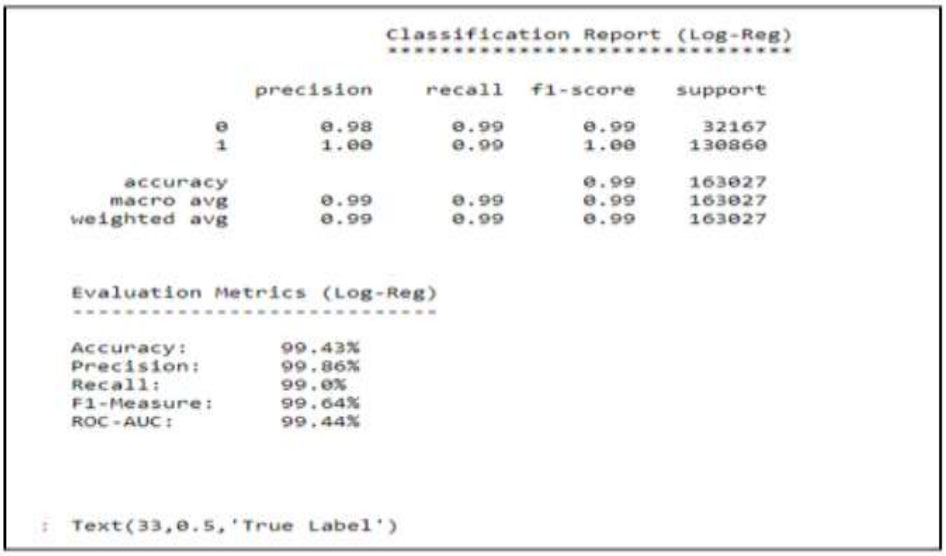

Figure 3: Model's Classification Report

4.6 Results Of Models Trained On (All Features) With Varying 'Gamma' Parameter Values

In the early section of this work, it was stated that OCSVM model algorithms have some parameters which are needed to be adjusted which are called hyperparameters, which help to find the balance between bias and variance and thus, prevent the model from overfitting or underfitting. To test for an optimal performance of the model several varied values of the gamma parameter were used to create some iteration of the classification into 8 different results groups given by g1 to g8 and comparing their equivalent result with that of the default hyperparameters. It was observed that the scale value of gamma used at g7 gives the best performance of the model with an accuracy of 95% compared to 90% got with the default parameters. Table 3 presents all the results of the model using varying values of gamma parameters and their equivalent performance metrics.

Figure 4 shows the Graphical representation of Results of Models Trained on (All Features) with varying 'gamma' parameter values for each of the 8 groups of varying gamma value with that of the default parameter. The ash colour is the most dominant group on the graph wish represents g7 with the highest values of the performance matrices.

4.7 Results (%) Of FAR Vs DR (All Features) With Varying 'Gamma' Parameter Values

Figure 5 shows the percentage False Alarm Rate against the Detection Rate (DR) using all the features of the dataset with varying gamma parameter values.

Table 3: Results of Models Trained on (All Features) with Varying 'gamma' parameter values

	default_params	g1	g2	g3	g4	g5	g6	g7	g8
Accuracy	90	90	90	90	90	87	85	95	80
Recall	100	100	100	100	100	100	100	97	100
Precision	89	89	89	89	89	87	84	97	80
f-score	94	94	94	94	94	93	91	97	89
roc-AUC	75	75	75	75	76	69	62	93	50

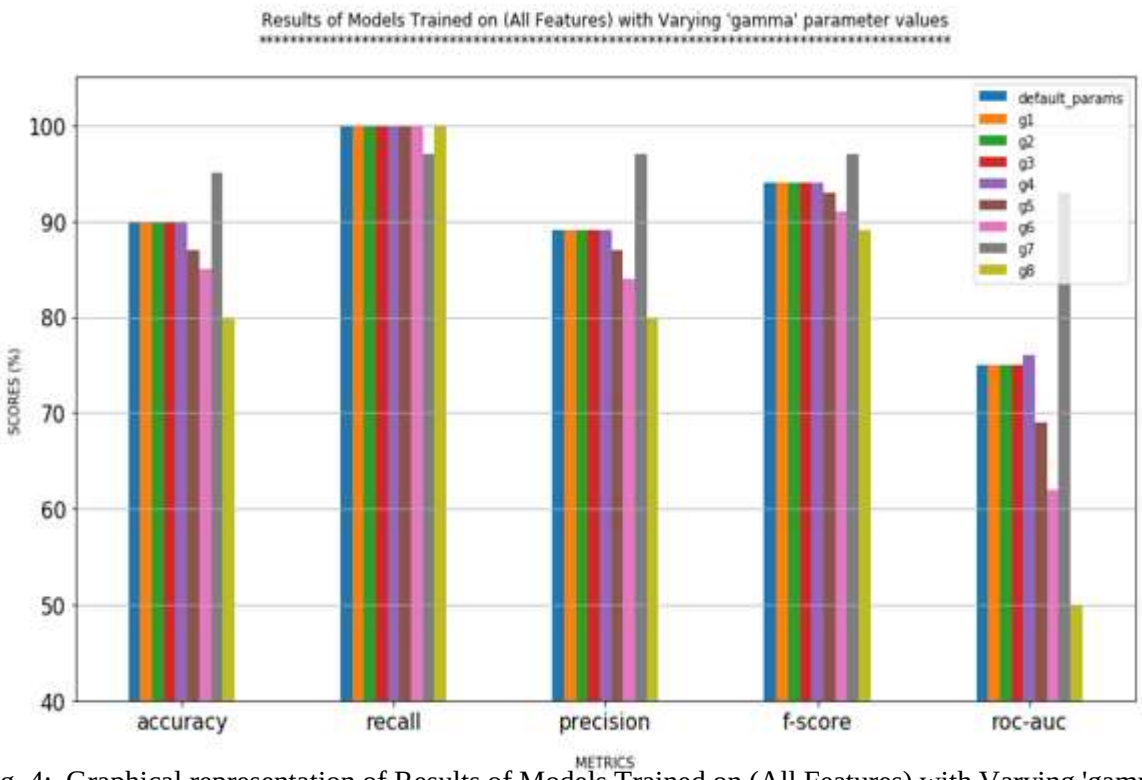

Fig. 4: Graphical representation of Results of Models Trained on (All Features) with Varying 'gamma' Parameter Values

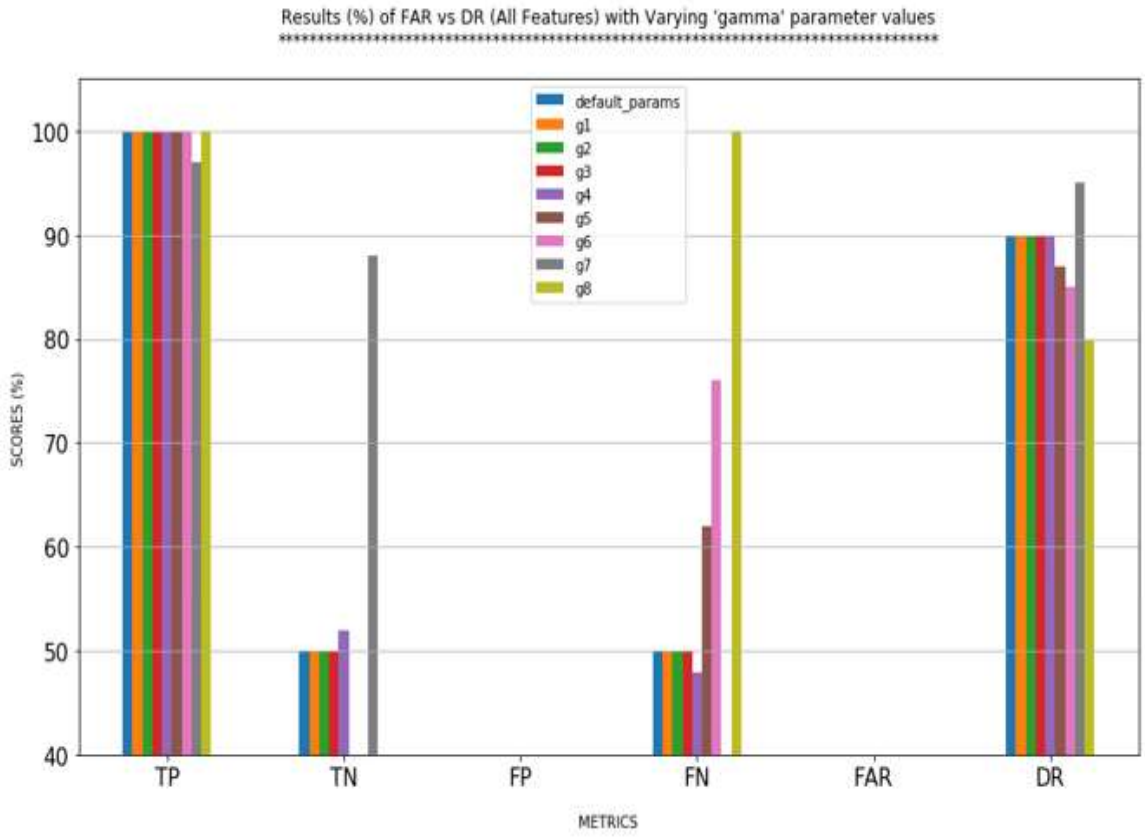

Figure 5: Graphical Representation of Results (%) of FAR vs DR (All Features) with Varying 'gamma' Parameter Values

5. CONCLUSION

A model for detection and classification of network anomaly was developed in this research work using a machine learning model. An Incremental approach was used to improve and enhance the network anomaly detection in a network to reduce false alarms with the One-Class Support Vector Machine (OCSVM) used to classify network anomalies in network traffic. The methodology used for the design was divided into stages which include dataset collection, Exploratory data analysis, Pre-processing, and finally the classification of the anomaly in network traffic. KDD '99 datasets were collected online, the data was explored (analyzed) and passed through the Pre-processing stages. The network anomaly was classified as normal and attack. Additionally, by using the One-Class Support Vector Machine (OCSVM) technique, network anomaly was predicted accurately at about 95% accuracy rates. This model has very great potential to be further improved in the future for further research.

References

[1] Balakrishnan, S., Venkatalakshmi, K., & Kannan, A. (2014). Intrusion detection system using feature selection and classification technique. *International journal of computer science and application*, *3*(4), 145-151.

[2] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2012). Survey on incremental approaches for network anomaly detection. *arXiv preprint arXiv:1211.4493*.

[3] Chakraborty, N. (2013). Intrusion detection system and intrusion prevention system: A comparative study. *International Journal of Computing and Business Research (IJCBR)*, *4*(2), 1-8

[4] Daya, B. (2013). Network security: History, importance, and future. *University of Florida Department of Electrical and Computer Engineering*, *4*.

[5] Joshi, J. B., Aref, W. G., Ghafoor, A., & Spafford, E. H. (2001). Security models for web-based applications. *Communications of the ACM*, *44*(2), 38-44.

[6] Karami, A. (2018). An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities. *Expert Systems with Applications*, *108*, 36-60.

[7] Liu, W., Luo, W., Lian, D., & Gao, S. (2018). Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 6536-6545.

[8] Lamrini, B., Gjini, A., Daudin, S., Pratmarty, P., Armando, F., & Travé-

[9] Mudzingwa, D., & Agrawal, R. (2012, March). A study of methodologies used in intrusion detection and prevention systems (IDPs). In *2012 Proceedings of IEEE Southeastcon*. 1-6. IEEE.

[10] Nitin, T., Singh, S. R., & Singh, P. G. (2012). Intrusion detection and prevention system (IDPs) technology-network behavior analysis system (nbas). *ISCA J. Engineering Sci*, *1*(1), 51-56.

[11] Samrin, R., & Vasumathi, D. (2017, December). Review on anomaly-based network intrusion detection system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)* 141-147. IEEE.