ISSN No. 0976-5697

Volume 8, No. 5, May – June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Comparison of Machine learning algorithms in Anomaly detection

Gunseerat Kaur
Student, Computer Science & Technology (cyber security)
Central University of Punjab
Bathinda, Punjab, India

Dr. Satwinder Singh
Astt. Prof., Computer Science & Technology
Central University of Punjab
Bathinda, Punjab, India

Abstract: Presence of threats in networks requires to strengthen the procedure of intrusion detection, with evolving threats, better threat recognition is required. In order to secure the networks and detect the attacks at various sub-levels, there is a keen interest in implementing an efficient machine learning methodology to seek the malignant from benign. Anomaly detection, supervised or unsupervised deals to handle the perturbations from the normal network, indicating faults defects and others malicious activities. This paper discusses the use of Support Vector Machine and multilayer perceptron to detect anomalies over network traffic.

Keywords: Machine learning, Support vector machine, Multilayer perceptron, Anomaly detection, Intrusion detection systems

I. INTRODUCTION

Increasing technology in daily life poses inexperienced users as vulnerable and fruitful targets for malicious purposes such as identity theft, man-in-the-middle attacks, DoS attacks, etc.[1]. Sensitive data is stored on devices for ease is often preyed upon by cyber attackers. Cyber-attacks are classified into two broad categories: Targeted and Massive attacks [2]. Initially, targeted ones only apply to a single user or organization and concerns with exploiting their information but on the other hand in the latter one; the goals are any and many internet users at mass. The planned execution is carried out with the help of zombies which act on commands, or heavy internet traffic abuse occurs.

Network traffic monitoring and managing has become the essential component, monitoring all the flow is a basic need. This is turn could provide better scenario on the common front and also would give an insight into the absurd field [3]. With an emphasis on the study of the difference in patterns, anomaly detections are done to find any change in normal executions over the network. The primary goal of anomaly detection is to target any event falling in outside of a particular predefined set of behaviors. Anomaly detection [4] programs assume that any intrusive event is a subset of unusual activity [5]. When new attacks appear the anomalies serve as a purpose to confirm that the change in patterns reflects something unusual. This would enable to draw a clearer line of divide amongst the two. A class of profiles are defined for the purpose to provide the system with criteria to outline; apart from this there should also be tolerance to some overhead upon normal behavior; for instance binary codes, network failures, error generations and exceptions.

Network anomalies rely on the analysis of network traffic and characterization of dynamic statistical properties of the traffic normality to accurately and quickly detect network anomalies. A whole new class of sampling schemes, aim to sight notable beneficial features for anomaly detection. With a view to finding points that deviate from their original behavior many techniques can be devised [6], grouped in as statistical, supervised or unsupervised. Different methods require separate segregation techniques. Introducing the concept, [7] proposed that any unwanted activities or disruptions in the functioning were threats. Hence a system should be opted to monitor these threats. Intrusion detection systems could be such systems that keep a careful check on the operations of processes, and if any threats like stored in its reference-based [8] store are detected, the alarms get triggered. The problem lies in detection of new and unsurfaced attacks that may cause severe damage, if not full protection, at least alarms can be generated in such situations to mitigate the damage occurring.

II. DETECTION TECHNIQUES

Detection scheme plays an important role in determining all the information about the attack. Earlier when IDS was introduced, it was only based on detecting the signatures of the attacking viruses or malicious.

1) Signature based detection

SBD or Signature based detection builds on the scanning of executables and other files to identify their computing signature. SBD considers finding a possible pattern, which is called a signature. These signatures are provided for free (like in the case of exploit database) or are paid (like an anti-malware software's repository). When any anti-virus software scans an executable, process or a network data packet; it generates the signature of that file simultaneously looking for a match in its database, if a match is found the executable is deemed to be infected [9].

2) Anomaly based detection

Behaviour based detection considers its detection methodology by working on its appearance, the attributes of an executable, process or a network packet. The categories are provided in which normal and abnormal classifications and values are submitted according to the official standards. These then help train the detector to understand the differences. Purely classification is done on a single class basis which then can be branched onto the lateral set of conclusions [10]. This technique is an advancement to signature-based technique, only enhancing the disadvantage of signature-based technique in which only known set of values could be detected. It is seen as a slight variation of behaviour-based detection, wherein analysis is done by studying the behavior of all files in the training phase.

III. RELATED WORK

Intrusion detection systems have been recognising malicious traffic from network traffic from provided set of rules. Provided with the measurements, signatures previously, IDS [7] works upon the principle of detecting the network traffic packets which violate the policies of the particular system. Cases include recognition of suspicious traffic and looking for signatures, most IDSs like Snort [11], Suricata [12] will find dangerous conditions that prejudice the rules provided in their database for classifications while segregating the network traffic. Inspecting the incoming and outgoing suspicious patterns requires three-fold procedure of firstly taking the dataset, then constructing an initial set of rules to draw the line between the two set of distinctions to be done(Northcutt Stephen, et al., 2005). Third and the last phase is to respond to the segregated traffic and mitigate measures and generate reports. These systems to categorize the network traffic can be based on two types of schema. Signature based scanners used the approach to scan out the likely patterns based on signatures stored either as rules or some virus definitions in the IDSs database. Signature-based methods showed success by providing more accuracy, time savings, and detailed log [13]. The problem arises in regular update for definitions in the rule sets, which with time might lead to sluggishness in the systems with not proper managing. Anomaly based detection technique is another of detection type, on which the next generation IDSs are based upon [3]. Exploring the theme in this project, these systems were proposed with the technique to detect irregularities in the behaviours of the packets in the network traffic. This technique uniquely identifies the behaviours of the network, measures the prior deployed thresholds.

From past two decades, detection techniques have been adopting machine learning techniques in their methods. Machine learning has been used in various fields such as fraud detection, web search altering, real-time advertisements, text analysis, pattern and image recognition, and much more. The main idea for Machine learning remains to recognise the patterns in the given set of data [14]; Machine Learning provides with algorithms that act as a basis to classify the patterns in the given set of data. This technique provides with the efficient basis for research in this context; the algorithms provide techniques for constructing the data mining rules [15].

Data mining can be a discovering method for information of various kinds from existing or supplied data, important is the manner of asking for information. The desired attributes are fed into algorithms to handle data and normalise it for scalable presentation with results. In different interactions, results vary and therefore the data mining methods crux out the best-matched scenarios[5]. These results can further be decisive with machine learning methods, on an instance, data mining can be also done with rules and not an algorithm. Data mining gets certain properties with algorithms; these classifications can be helpful. These detections and classifications were firstly given by [16]. In intrusion detection, these terms were categorized as knowledge based and behaviour-based intrusion detection. Knowledge-based detection attempts to look for patterns similar to the ones stored in its knowledge base by the knowledge engineer. These knowledge bases impart actions on how the system will configure on classification[17].

Anomaly detections scheme has had its relative better chances over the signature based scheme, exploring this very property. Many methods to enhance its recognition for anomalies were proposed. Flow control, level hierarchy in network, unsupervised methods, supervised methods, automatic feature selection or extraction of details at tiers. This approach has large variations to its credibility. These variations encapsulate different algorithms and suggestions to combat better accuracy and less time taking process for classifiers to classify the incoming network traffic. Taking snippets from this survey, in this project using the machine learning method, anomaly detection will be performed over the two machine learning algorithms, SVM and MLP.

IV. DATA COLLECTION

Dataset selection is a task to determine what type of dataset is to be fed to the classifier in order to achieve the optimal results*; i.e*. better accuracy, less time and resource consumption. With a view to summon an anomaly detection based system, there is a requirement to explore the standard datasets that provide the basis for learning [18]. The KDD cup99[19] data set has been used for a long time been considered to be an excellent data set when training algorithms. One of the reasons for that is due to it great amount entries and that it is labelled traffic. This is one of the premises for using supervised learning that the data is labelled and the machine actually knows what kind of threats it should recognise.

The NSL-KDD[20] dataset is an improvement of the old KDD CUP99[19] data set and will be used in this study. It has been improved by its creators by removing some of the redundant data points that could cause errors and give better results than what should be [21]. The KDD cup 99, though defined the attacks well, contained redundant data records which affected the classification records. The NSL-KDD is supposedly more exact in the way it is created and addresses most of the issues mentioned in the criticism of the original dataset. After the dataset is chosen, many tasks are required to be performed on it for making it suitable for the classifier to read.

1) Data Acquisition

Retrieving dataset from a particular repository is the process of data acquisition, in this case, two types of data were acquired. Firstly the NSL-KDD set which was downloaded directly from its website[1]. Secondly the creation for testing dataset required to set up a network of virtual machines. In this created network, multiple attacks similar to the downloaded dataset were performed and values were logged from the network.

[1] *http://www.unb.ca/research/iscx/dataset/index.html*

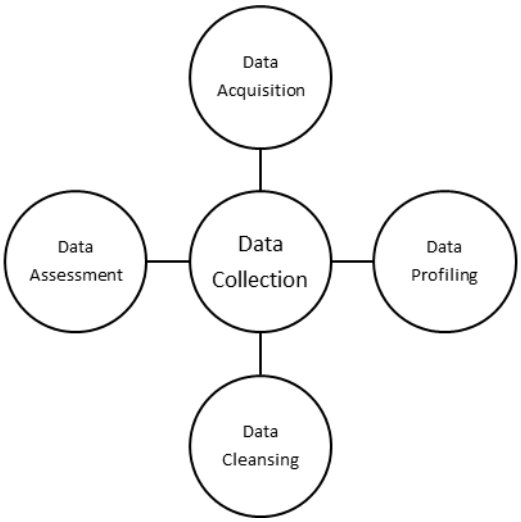

Fig 1 Data Preparation Process

2) Data Assessment

After obtaining the dataset, it can't be directly fed into next process. Data assessment is the process of evaluating data in order to determine whether they meet the quality required for processes and are of the right type and quantity to be able to actually support their intended use [22]. Particularly, integrity is preserved in maintaining data by assessing it. Data assessment ensures quality assurance standards, concerns.

3) Data Profiling

Data profiling can be seen as a process of examining the data available in the dataset or files, figuring statistics and summaries in relation to the data. Metadata, resources of dataset, value patterns, key-structure and functional dependencies are determined in this.[18].

TABLE 1 DATA PROFILING AND ASSESSMENT FOR DATASETS

summary	mean	Std dev	min	max
land	1.984	0.014	0	1
logged_in	0.397	0.489	0	1
root_shell	0.001	0.036	0	1
su_attempted	0.001	0.045	0	2
is_host_login	7.938	0.002	0	1
is_guest_login	0.009	0.096	0	1

4) Data Cleansing

Data cleansing, cleansing or scrubbing, refers to the procedure of removing the unwanted values according to purpose or usage of data. It is the process of removing corrupt, inaccurate parts of dataset that will only hinder in optimal classification. It is important to identify these foreign parts by either replacing, deleting or modifying them from data.

V. MACHINE LEARNING

1) Support Vector Machine

Support vector machines or SVMs belong to the categories of regression and classification. An example in machine learning algorithms these are capable of very high dimensional tasks and are a collection of machine learning algorithms that can be used to recognise patterns in a given dataset [23]. Given a set of training data, we would like to classify new examples into one of the possible two categories. For achieving such a task SVM training algorithm can be used to build a model which is capable of performing such classifications that separate different classes by a hyperplane. Omitting the details of the calculation, there is a single fundamental property emphasised on. Apparently, both quadratic programming and final decision depend on the dot product of common patterns classified. This kind of classification allows a linear kernel to act as a non-linear kernel and optimise the solution. As shown in Eqn 3:

$$t(w,\xi) = \frac{1}{2}\|w^2\| + \frac{c}{m}\sum_{i=1}^{m}\xi i \quad \ldots (1)$$

m = number of training patterns …(2)

$\xi i = \pm 1$ …(3)

There are non-zero coefficients which meet at P(xi, yi) meet conditions of the support vectors; The other coefficients are maximised in turn. Cost parameter is denoted by C in Eqn 1,2,3; this is the measure of the misclassification done by the algorithm in deciding the measures while classifying.

2) Multilayer perceptron

Artificial neural networks or MLPs combines an approach that involves layering, the three layers: input layer, hidden layer and output layer. A feed forward type of network, this develops a directed graph with first layer fully connected to the next layer [24]. Each node takes an input from its neighbor, passing it on to next and generating a final output signal. Along with this processing, there is adjustment of weights. Weights in MLPs refer to the bond between the nodes, how much of an affect will take place on another node with respect to previous node's output. A structure similar to parallel processing system, the nodes are placed next to each other in a similar manner with units in same level carrying out their computations at the same time. Fig 2 shows the general model of an MLP

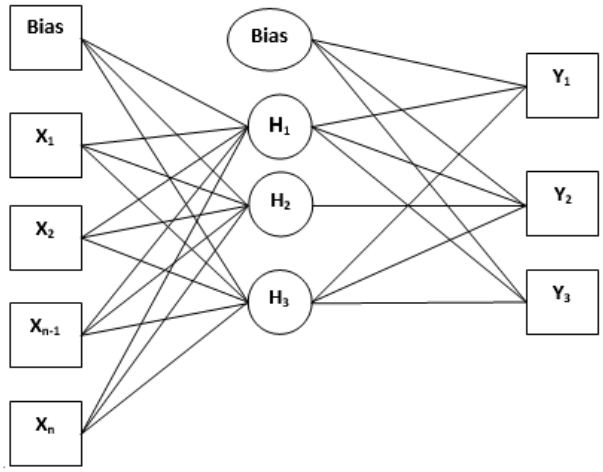

Fig 2 Multilayer perceptron's description

VI. OBSERVATIONS TO BE RECORDED

The parameters of classifiers will be then analyzed. Following are the classifiers' parameters that will be analyzed:

1) ***True Positive:*** *A true positive test result is one that detects the anomaly when the anomaly is present.*

2) ***True Negative:*** *A true negative test result is one that does not detect the anomaly when the anomaly is absent.*

3) ***False Positive:*** *A false positive test result is one that detects the anomaly when the anomaly is absent.*

4) ***False Negative:*** *A false negative test result is one that does not detect the anomaly when the anomaly is present.*

TABLE 2 OBSERVATIONS

Anomalies→ Test ↓	**Present**	**Absent**
Positive	True Positive	False Positive
Negative	False Negative	False Positive

These parameters form the confusion matrix further which entails to produce the following set of measures that will help in defining performances for classifiers over different attack categories.

5) ***Accuracy:***

Accuracy measure is the proportion of predicted anomaly prone modules/ packets that are inspected out of all modules. Accuracy includes both trueness and precision, which is an overall for both. It is defined as the closeness of obtained results with the actual results and precision is reoccurrence of the results.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

VII. RESULTS AND DISCUSSION

1) ***Performing binary classification***

Binary classification is performed in order to compare the accuracy, time consumption and usage of resources. It is a method of analyzing how well binary classification works in simple cases like binary segregation is. In this experiment the dataset's complete instances were visualized rather than distributing it with respect to some criteria. The two classes in this experiment are anomaly class out of which predicted and actual are compared and in Normal class in which predicted and actual no. of packets are compared in Fig 4 for MLP and Fig 5 for SVM, respectively. Multilayer perceptron was implemented on the simplest epochs, these epochs produced a fair set of results on the binary classification as shown in Fig 4 The epochs were chosen according to the schema of retrieving best results, as shown in table 4, the learning rate was best at 0.3 and it was noticed that larger the dataset, smaller the rate of learning.

TABLE 3 MLP SETTINGS FOR PERFORMING OPTIMAL CLASSIFICATION

Parameter	**Value**	**Information**
Learning rate	0.3	Determines the learning capability after regular intervals
No of epochs	1000	One forward, one backward counts as a single epoch, determines the batch size for MLP
Momentum	0.2	Responsible for adding a fraction of weight to the next node

After the variables and their values are determined, he test dataset is fed into the classifier after pre-processing. Fig 4 shows the resultant classification scores. Upon classifying, in binary classification over the four types of attack categories the attacks were classified with the efficiency. Out of supplied 20405 anomaly contained test packets, 15668 packets were classified correctly by MLP. Similarly, for Normal packets fed into classifier, 22304 packets, 17496 packets were correctly classified.

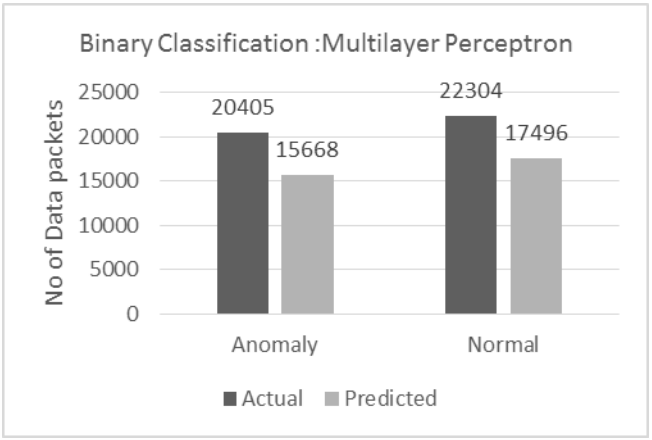

Figure 4 Binary Classification: Multilayer Perceptron

Reaching to a classification accuracy of 76.7%, MLP shows good performance in classifying anomaly packets in the total supplied no. of packets. On the other end, for normal class it shows an accuracy of 78.4%. Performing the classification multiple times it was observed that the results at these thresholds were better than other values supplied.

SVM implementation as a second algorithm in this project, used another set of thresholds as shown in table 4, these values showed the best results as compared to rest of the values. As a binary classifier, SVM used a cache size of 40 MB which led to an optimum kernel for running these many instances. The cost parameter was kept at 1, avoiding misclassification of training examples.

TABLE 4 SVM THRESHOLDS FOR OPTIMAL RESULTS

Parameter	**Value**	**Information**
cache_size	40	Cache size in memory(default 40MB)
cost c parameter	1	Cost of constraint violation
epsilon, ζ	0.001	Epsilon determines the accuracy of the approximated function

Upon classifying, in binary classification over the four types of attack categories the attacks were classified with the efficiency. Out of supplied 20405 anomaly contained test packets, 19249 packets were classified correctly by SVM, which elaborately performed better than MLP. Similarly, for Normal packets fed into classifier, 22304, 18993 were correctly classified.

Showing an accuracy in detecting anomaly packets at 86.30% and 93% for normal classification. Clearly SVM performed slight better in terms of binary classification, this may be explained as the resultant of having fewer classes to

work with. On an initial level, SVM are elaborately faster in the two class classifications, a model built predicts the examples falling into one of which classes, settling the gap of hyperplane between the distributions at a permissible optimum
limit.

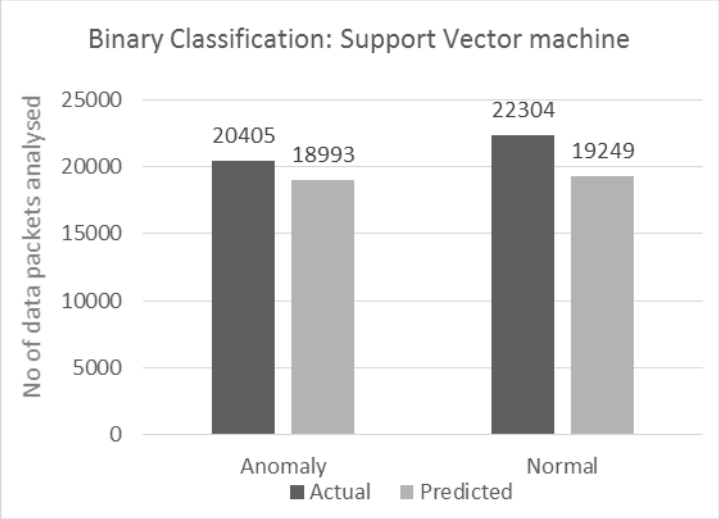

Figure 5 Binary Classification on Support Vector Machine

2) ***Performing multi-class classification using SVM and MLP***

The previous experiment uses the algorithm SVM and MLP to perform a dual-class classification. The algorithms were applied previously, exempting the measure of time and resource usage. Supervised learning explains that in learning what the attack is and how it learns by grouping the different attacks and recognizes the patterns from the group of attacks. In this experiment the results are categorized into different attack-learning patterns, these attacks were chosen according to the parent attack category. Collectively equal no. of instances were supplied for this experiment to the classifier. This was done in order to detect the anomalies in a certain set of attacks. These attacks are then named as different classes for classification on a multi-level. From the training set, different categories were generated which denoted different results for the test dataset. SVM and MLP were required to perform multi-class classifications and their accuracy for singular and combined correct classification has been derived.

In MLPs implementation, there were up to 43 sigmoid nodes generated while performing the multiple class classification. Applied to the equivalent thresholds as in the above case, the MLP's average correctly classification resulted up to 87.98 % which in particular was best for Neptune attack detection. Evidently from the confusion matrix displayed in table 4 shows that rate of detection in Neptune attack, detection of SYN flood packets on a TCP/IP implementation scored the highest of 95.40% while the detection of ICMP ping packets for Ipsweep were detected on the lowest at 88.95%.

TABLE 4 CONFUSION MATRIX IN PERCENTAGES OF MULTIPLE ATTACK CLASSIFICATION

Predicted→ Actual ↓	**Rootkit %**	**Smurf %**	**Neptune %**	**Land %**	**Ipsweep %**	**Nmap %**	**Dictio %**
Rootkit	**91.65**	2.35	1.30	0.90	1.20	1.30	0.80
Smurf	1.10	**89.30**	3.60	1.90	0.85	1.70	0.85
Neptune	0.10	0.80	**95.40**	0.75	2.15	0.05	0.10
Land	0.60	0.63	4.85	**90.25**	1.25	1.27	0.75
Ipsweep	0.00	0.70	4.40	5.10	**88.95**	0.65	0.00
Nmap	0.65	0.95	2.35	1.20	1.15	**92.70**	0.60
Dictionary	0.00	0.95	0.70	0.88	1.25	1.62	**93.60**
Buffer overflow	1.80	1.95	1.10	0.10	1.35	0.85	0.85
Average correctly classified instances: 87.98%							

On similar lines of classifying the multiple classes in SVM, it was initiated by a binary classification into anomalies and then supplying set features for further attack classifications. Also since SVMs support many different kinds of kernel, it is important to seek what adjustments to the values produces optimal results with respect to the classification being performed. The problem with such implementation is that the noise increases with many binary classifiers gearing up for a common multi class classification. So in order to maintain the accuracy-efficiency trade off, table 5 enlists the set of values required to efficiently devise a radial-basis kernel for functioning of SVM.

TABLE 5 MULTICLASS VALUES FOR SVM-RADIAL BASIS KERNEL

Parameter	**Value**	**Information**
Gamma	0.1	Reverse for standard deviation, used for determining similarity between two points.
cost c parameter	10	Cost of constraint violation

Performing these parameter optimizations table 5 shows the resultant accuracy measures derived for different classes in SVM.

TABLE 6 ACCURACIES FOR SVM

Predicted→ Actual ↓	**Rootkit %**	**Smurf %**	**Neptune %**	**Land %**	**Ipsweep %**	**Nmap %**	**Dictionary %**	**Buffer overflow %**
Rootkit	**93.80**	0.20	1.30	2.30	0.95	0.85	0.30	0.20
Smurf	0.40	**94.60**	0.60	0.70	1.10	0.80	0.90	0.90
Neptune	0.50	0.30	**96.20**	0.80	0.90	0.60	0.20	0.40
Land	0.20	0.10	3.80	**91.50**	0.90	2.50	0.80	0.70
Ipsweep	0.90	0.90	9.20	0.85	**84.30**	1.25	1.10	2.50
Nmap	0.90	0.90	9.20	2.30	1.50	**83.20**	0.80	1.20
Dictionary	0.80	0.90	2.10	0.10	0.10	7.20	**88.70**	0.10
Buffer overflow	0.45	0.80	0.90	1.20	0.80	1.20	1.10	**92.85**
Average correctly classified instances: 90.64%								

From the above table 5, clearly highest accuracy for detection is shown for Land attacks, recognized for its same source and destination address, it is accurately classified at 96.20%. On the contrary the lowest detection accuracy is shown in Dictionary attack, done on the basis of number of failed login attempts, at 83.20%.

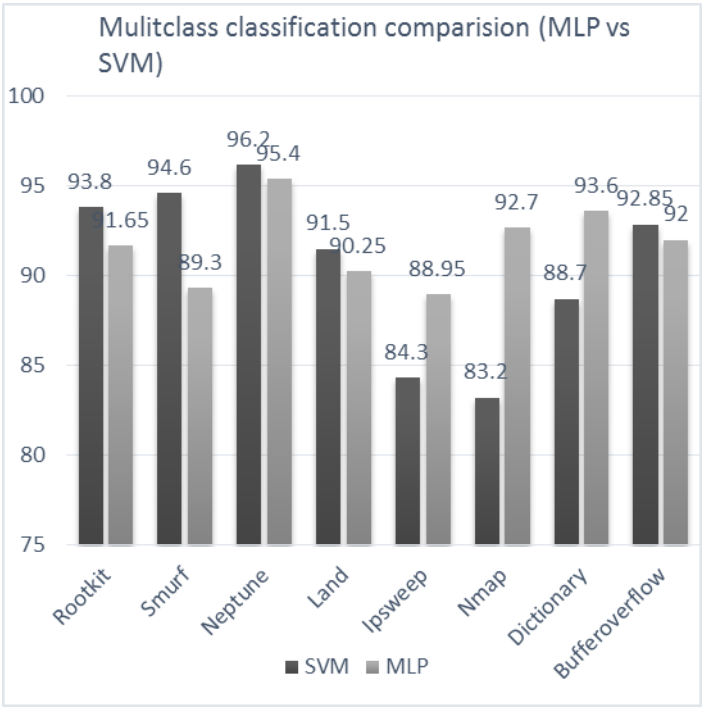

FIG 7 MULTICLASS COMPARISON BETWEEN CLASSIFIERS

With an average correctly classified instances of up to 90.64%, Support vector machine show an evident higher accuracy in predicting the anomalies in different types of attacks as compared to multilayer perceptron which show a slightly lower performance than SVMs with an overall accuracy of correctly classified instances at 87.98 %.

VIII.CONCLUSION AND FUTURE WORK

The main goal of this paper is to implement and compare the results for two machine learning algorithms in performing anomaly detection. The study has utilized a machine learning technique called supervised learning. It was implemented through usage of SVM & MLP which are supervised classification algorithms. The results gained from implementing were then compared in order to see if of them is best suited to perform anomaly detection in a network environment. The network environment was simulated by using a dataset containing samples of network traffic. The data set contains different indicators for multiple attacks blended in with normal traffic. There are labels linked to each sample, which makes it possible for the algorithms to differentiate between patterns of attacks and normal traffic. The algorithm is implemented using both binary and multiclass data. This was done in order to observe how well the algorithms performed, when exposed to both binary and multiclass data. The experiments conducted in this paper were implemented in order to present solutions in detection of anomalies using optimal parameters. For future implementations, the models created in this paper shall be tested on real time traffic and the variances will be checked

IX. REFERENCES

[1] L. Bilge, T. Strufe, D. Balzarotti, E. Kirda, and S. Antipolis, “All Your Contacts Are Belong to Us□: Automated Identity Theft Attacks on Social Networks,” in Proceedings of the 18th international conference on World Wide Web, 2009, pp. 551–560.

[2] T. Rid and B. E. N. Buchanan, “Attributing Cyber Attacks,” J. Strateg. Stud., vol. 0, no. 0, pp. 1–34, 2014.

[3] R. Kandhari, V. Chandola, A. Banerjee, V. Kumar, and R. Kandhari, “Anomaly detection,” ACM Comput. Surv., vol. 41, no. 3, pp. 1–6, 2009.

[4] M. V. Mahoney and P. K. Chan, “An Analysis of the 1999 DARPA / Lincoln Laboratory Evaluation Data for Network Anomaly Detection,” Recent Adv. Intrusion Detect., pp. 220–237, 2003.

[5] K. Giotis, G. Androulidakis, and V. Maglaris, “A scalable anomaly detection and mitigation architecture for legacy networks via an OpenFlow middlebox,” Secur. Commun. Networks, vol. 9, no. 13, pp. 1958–1970, 2016.

[6] M. A. Ferrag and A. Ahmim, Security Solutions and Applied Cryptography in Smart Grid Communications. 2017.

[7] D. E. Denning, “An Intrusion-Detection Model,” IEEE Trans. Softw. Eng., vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[8] Y. Pei, O. R. Zaïane, and Y. Gao, “An efficient reference-based approach to outlier detection in large datasets,” in Proceedings - IEEE International Conference on Data Mining, ICDM, 2006, pp. 478–487.

[9] Y. Zeng, K. G. Shin, and X. Hu, “Design of SMS commanded-and-controlled and P2P-structured mobile botnets,” in Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks - WISEC ’12, 2012, p. 137.

[10] R. R. Northcutt Stephen, Zeltser Lenny, Winters Scott, Kent Karen, Inside Network Perimeter, vol. 1. 2005.

[11] M. Roesch, “Snort: Lightweight Intrusion Detection for Networks.,” LISA ’99 13th Syst. Adm. Conf., pp. 229–238, 1999.

[12] Suricata, “Suricata Open Source IDS / IPS / NSM engine,” 2015. [Online]. Available: https://suricata-ids.org/. [Accessed: 12-May-2017].

[13] R. Sekar et al., “Specification-based anomaly detection: a new approach for detecting network intrusions,” Proc. 9th ACM Conf. Comput. Commun. Secur., vol. 26, no. 2, pp. 265–274, 2002.

[14] S. Buthpitiya, “Modeling Mobile User Behavior for Anomaly Detection,” 2014.

[15] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, “Towards effective feature selection in machine learning-based botnet detection approaches,” 2014 IEEE Conf. Commun. Netw. Secur. CNS 2014, pp. 247–255, 2014.

[16] A. Jayasimhan and J. Gadge, “Anomaly Detection using a Clustering Technique,” Int. J. Appl. Inf. Syst., vol. 2, no. 8, pp. 5–9, 2012.

[17] P. Gogoi, D. K. Bhattacharyya, and J. K. Kalita, “A rough set-based effective rule generation method for classification with an application in intrusion detection,” Int. J. Secur. Networks, vol. 8, no. 2, p. 61, 2013.

[18] S. Omar and H. H. Jebur, “Machine Learning Techniques for Anomaly Detection□: An Overview,” Int. J. Comput. Appl., vol. 79, no. 2, pp. 33–41, 2013.

[19] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” IEEE Symp. Comput. Intell. Secur. Def. Appl. CISDA 2009, no. Cisda, pp. 1–6, 2009.

[20] H. Chae, B. Jo, S. Choi, and T. Park, “Feature Selection for Intrusion Detection using NSL-KDD,” Recent Adv. Comput. Sci. 20132, pp. 184–187, 2013.

[21] N. Görnitz and K. Rieck, “Toward Supervised Anomaly Detection,” J. Artif. Intell. Res., vol. 46, no. 4, pp. 235–262, 2013.

[22] P. Evans, "Scaling and assessment of data quality," in Acta Crystallographica Section D: Biological Crystallography, 2006, vol. 62, no. 1, pp. 72–82.

[23] D. Meyer, "Support Vector Machines," cran-project.org, 2015. .

[24] R. Tadeusiewicz, "Neural networks: A comprehensive foundation," Control Eng. Pract., vol. 3, no. 5, pp. 746–747, 1995.