ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

Network Intrusion Prevention System Using Machine Learning Techniques

Chanakya G*, Kunal P, Sumedh S, Priyanka W, Mahalle PN

Smt. Kashibai Navale College of Engineering Pune, India

Abstract: Secured data communication over networks is always under threat of intrusions and misuses. A Network Intrusion Prevention and Detection System (IPDS) is a valuable tool for the defense-in-depth of computer networks. Network IPDS look for known or potential malicious activities in network traffic and raise an alarm whenever a suspicious activity is detected. The Intrusion Detection Systems most commonly used in enterprise networks are signature-based, because they can efficiently detect known attacks while generating a relatively low number of false positives. Anomaly-based detection systems usually produce a relatively higher number of false positives, compared to the misuse-based or signature-based detection systems because only a fraction of the anomalous traffic is derived from intrusion attempts. As a matter of fact, it has been shown that the false positive rate is the true limiting factor for the performance of IDS, and that in order to substantially increase the Bayesian detection rate, P (Intrusion |Alarm), the IDS must have a very low false positive rate. One-class classification algorithms pursue concept learning in absence of counter examples, and have been shown to be promising for network anomaly detection. This project aims to use one-class classifier that is One-Class Support Vector Machines to detect network attacks that bear form of port-scan attacks for very low false positive rates.

Keywords: Intrusion detection systems; Intrusion prevention system; Support vector machine; One-class support vector machine; One-class classification

I. INTRODUCTION

The rise of globalization has encouraged multi-location organizations to adopt the use of enterprise networks. The key purpose of this network is to eliminate isolation of branches or users while maintaining satisfactory performance, reliability, and security. This system may include implementation of local area or wide area networks according to the organization's needs. Similarly, the organization can also integrate systems such as Windows, Linux or Macintosh operating systems. Thus, enterprise networks can be defined as an organization's communication channel that helps connect users across departments, cities or even countries to facilitate data accessibility. The advantage of enterprise networks is that it reduces communication protocols and improves internal and external enterprise data management. However, these advantages come at a price: the risk of network attacks, some of which may be fatal to the enterprise. Enterprise network security should be a high priority when considering setup as the growing threat of hackers trying to infect as many computers possible is increasing exponentially. For corporations, security is important to prevent any intruder from breaching into their systems. The purpose of network security is essentially to prevent any harm to the company, through misuse of data. We may come across a number of problems if network security is not implemented properly. Some of these are breaches of confidentiality, data destruction, and data manipulation. Intrusion Detection Systems (IDS) are valuable tools for the defence of computer networks. Network IPS look for known or potential malicious activities in network traffic and raise an alarm while preventing the attack whenever a suspicious activity is detected. Two approaches to intrusion detection are signature and anomaly detection. Signature detection is based on a set of known malicious activities. This particular set contains a set of rules referred as attack signatures. Activities that match an attack signature are classified as malicious. Whereas, anomaly detectors are based on a description of normal activities. As malicious traffic is expected to be different from normal traffic, a suitable distance measure allows anomaly-based IDS to detect attack traffic. The most commonly used IDS in real networks are signature-based because the false positive generation rate is relative. Whereas, Anomaly-based detection have a very high false positive generation rate because only a fraction of anomalous traffic actually are intrusion attempts. Nevertheless, an anomaly detector are able to detect zero-day attacks, whereas signature-based systems are not as it is very difficult and expensive to obtain a labelled dataset that is representative of real network activities and contains both normal and attack traffic,

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

unsupervised learning approaches for network anomaly detection have been recently proposed. These methods aim to work on datasets of traffic extracted from real networks without the necessity of a labelling process. In Unlabelled anomaly detection systems, we assume that the likeliness of detecting attack patterns in the extracted traffic traces is usually much lower than the likeliness of normal pattern detection. Furthermore, we can use signature-based IDS in order to filter the extracted traffic by removing the known attacks, thus further reducing the number of attack patterns possibly present in the dataset. One-class classification algorithms pursue concept learning in absence of counter examples and have been shown to be promising for network anomaly detection. Through this project, our aim is to use one-class classifiers to detect zero-day attacks with respect to probing attacks and prevent them using an Intrusion Prevention System with minimum false positives and true negatives along with minimum latency.

II. LITERATURE SURVEY

There was a need to evaluate ways in which you can reduce the number of false positives and false negatives detected by the IPDS, for this purpose a detailed literature survey was done which included going through various research and survey papers on IPDS. Thus, we came across various papers which provided a part of the solution. Ying-Dar L, et al. Tai have explained the concept of credibility based weighted voting In this paper [1] they have employed multiple IDS and applied a method called CWV to the outcomes of multiple IDS for reducing FPs and FNs. First the CWV scheme evaluates the creditability of each individual IDS. For each IDS the scheme then assigns different weights to each intrusion type according to its FP and FN ratios. Later the outcomes of each IDS are merged using a weighted voting scheme. According to S Wu and E Yen data mining can be used for intrusion detection [2] they have considered four attack type: U2R, Probe, DOS, R2L and compared their accuracy, the detection rate and false alarm rate [3]. They used C4.5 and SVM algorithms to provide an accurate comparison of above four kinds of attacks. C4.5 acts better for probe, U2R and DOS attack, but SVM proved better in detecting false alarms, similarly [4]. Anuar NB, et al. has used a hybrid data mining technique along with decision trees. The writers have used training data set of KDD Cup99 and proposed a method for detecting and statistical analysis of both attack and normal traffics. A hybrid statistical technique based on data mining and decision tree classification has been used. These methods differentiate between false positives and attacks and thus reduce the misclassification. They prove the importance of decision trees for designing intrusion detection for class of DOS, normal and R2L by comparing decision tree algorithms and rule-based algorithms [5]. "False Positives Reduction via Intrusion Alert Quality Framework", written by Bakar NA, et al. In this paper the writers actualize an interruption ready quality system, to decrease false positive cautions in IDS, they advance every alarm with quality parameters, for example, rightness, precision, dependability, and affectability. Using these parameters helps in deciding the quality of the alert and whether it really is anything abnormal or just a false alarm. Al-Mamory SO and H Zhang in [6] "A Survey on IDS Alerts Processing Techniques" has provided a brief understanding of alert processing techniques [7]. It is a survey paper on the various alert processing techniques, they have described a data mining alert clustering strategy that gathers alerts whose primary causes can be compared and discover generalized alerts which assist the administrator or analyst to write filters [8]. Benjamin M, et al. in the paper "M2D2: A Formal Data Model for IDS Alert Correlation" have explained alert processing in a different way, Benjamin Morin proposed correlation of Information related to the characteristics of the monitored information system, information about the vulnerabilities, information about the security tools used for the monitoring the events [9]. In the paper "An Intelligent Intrusion Detection and Response System Using Network Quarantine Channels: Adaptive Policies and Alert Filters" Emmanuel Hooper has proposed a model to reduce false positives using adaptive responses of firewall rule sets on "network quarantine channels (NQC) using firewall architectures. The model is a combination of firewall architecture associated with response rules, to deny access to critical segments to suspicious hosts in the network. Author Kai Hwang, et al. "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes" has proposed [10] a hybrid model of signature based IDS and Anomaly Detection System to get low false-positive and to sense unfamiliar attacks [11]. The ADS was trained by exposing to abnormal traffic incidents from Internet connection, which detected anomalies more than the original two independent models, which reduced the total false positives and negatives generated. Byoungkoo K, et al. has proposed a method for high performance intrusion detection [12]. In this paper the authors propose the FPAG-based intrusion detection technique to detect and respond variant attacks on high-speed links. It is possible through novel pattern matching mechanism and heuristic analysis mechanism that is processed on FPGA-based reconfiguring hardware. The technique is a part of a proposed system, called ATPS (Adaptive Threat Prevention System) recently developed. That is, the proposed system has hardware architecture that can be capable of provide the high-performance detection mechanism [13]. Jose G, et al. have used techniques called

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

shunting. Shunting is a technique which provides significant benefits for network intrusion prevention in environments for which an IPS can dynamically designate portions of traffic stream as not requiring further analysis. Author Yaron W, et al. in their paper [14] has used a pattern-matching algorithm. The algorithm uses the concept of Ternary Content Addressable Memory and is capable enough of matching multiple patterns in a single operation. This algorithm is able to detect abnormalities much faster than most of the current detection methods, while attaining similar accuracy of detection.

2.1 Review of FRR and FAR

From all the studies it is concluded that while some papers proposed different approaches the majority have reduced the false positive in the same way. Correlating alerts improve the efficiency of the IDS. It not only decreases the false positive rate, but also improves the knowledge of attacks on the network. Despite the reduction of false positive, the methods still need to be improved as they still have weak spots [15].

In case for false negatives, anomaly detection does tend to produce more false negatives than in signature based detection. Therefore reduction of false negatives depends on the machine learning technique that you use and on how precise your training data is. From this literature survey we can conclude that, practically, it is not possible to build a completely secure system and false positive or false negative generation will continue. It is only possible to reduce the number of false positives or false negatives [16].

2.2 Review of Latency

To decrease latency is the same as increasing its throughput or performance. To implement a high-performance Intrusion Prevention and Detection System we need special hardware like high computing GPUs and CPUs, thus making it very costly [17]. Not many methods have been developed which reduce this cost to make it feasible to develop a high performance IPDS. Also, all methods are hardware based, there are extremely few cases where software based techniques are used in increasing performance of an IPDS.

III. ANOMALY BASED DETECTION

Anomaly based detection techniques scan through the entire network traffic and classify it as normal or anomalous. For classification you need to develop a training set, the system will refer this training set and classify the data as normal or anomalous, therefore the efficiency of your detection system depends on how well you develop your training set. The training set will define what normal traffic is and if the system comes across anything that is not in accordance with the training set it will be classified as anomalous. Because of this anomaly based detection techniques also have a high false positive generation rate. One major advantage of anomaly based detection techniques over signature based is the detection of zero day attacks since novel attacks are detected as soon as they take place [18]. There are various anomaly based detection techniques that are used like-Statistical Models, Cognition Models, Cognition Based Detection Techniques, Machine learning based detection techniques. In this paper we will be focusing on Support Vector Machine (SVM) which is a machine learning based detection technique. We will be inspecting network packets for probing attacks.

IV. PROPOSED SYSTEM ARCHITECTURE

From all the papers we referred to during our research phase, we came across different ways to implement IPS and IDS, but we did not come across a system where both IPS and IDS were implemented together as a cohesive unit. In our proposed architecture, IPS is in-line with the network and the IDS will sniff all the packets and will maintain a log to enter all entries of packets that go through from the internet into the internal network. Here, IPS and IDS will have different functionalities [19]. The payload of the packets coming in will be inspected by the IPS for any malicious code, and will either block the packet or accept it and allow it to flow into the internal network. If it comes across a packet which is unknown to the IPS it will block (Figure 1). The blocked packets will then be inspected by the IDS to look for more intricate details. We will be using one-class SVM as a machine learning technique to classify the packets as malicious or normal [20]. If the payload seems normal to the IDS then it will be allowed to flow back into the internal network along with the other packets. But if any irregularity is detected, the packet will immediately be discarded. After these operations have been performed by the IDS it will update its rule base and subsequently will send these updated rules to the IPS so that the IPS will know what to do if it comes across a packet of similar type. This process will be repeated whenever the IPS comes across an unknown packet [21].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

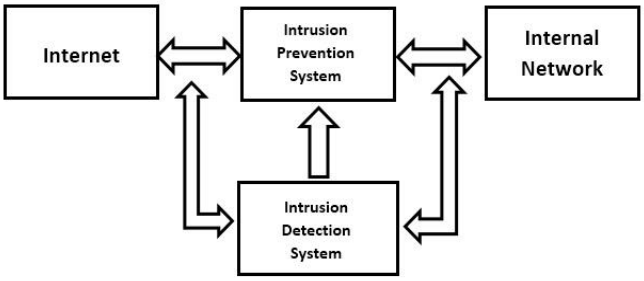

Figure 1: System architecture.

V. LEARNING SCHEMES

For various applications, which involve machine learning different learning schemes are implemented. For intrusion detection, two learning schemes are prevalent: classification and anomaly detection (Figure 2). Classification is the problem of identifying the class a new data object belongs to when given a training set of data containing observations whose category membership is known [22]. In other words, in classification, we have observations from two classes of data and the machine trains according to these observations. When a new observation is given as an input, the machine will classify the new observation as one of the two classes it has learned about. For this, the prerequisite is having two classes of data for training purposes [23]. Many a times, it is not possible to obtain sufficient amount of data for both the classes, and hence this method should be avoided if possible. For the reason given above, we adopt the anomaly detection learning scheme (Figure 3). Anomaly detection is the identification of the observations that do not follow a certain expected pattern (outliers) in a dataset. Here, only one class of datasets is required to train the machine. The focus here is on only one prominent class and on learning its structure. As a result, it is possible to differentiate that class from everything else (outliers). By using this approach, it is possible to detect unknown attacks [24].

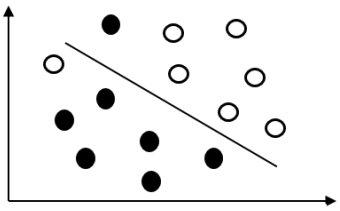

Figure 2: Classification.

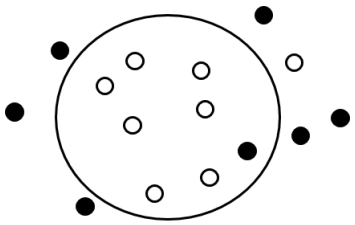

Figure 3: Anomaly detection.

VI. SUPPORT VECTOR MACHINE

We will briefly discuss about SVM before moving on to one-class classification and one-class SVMs. Support Vector Machines (SVM) is a discriminative classifier formally defined by a separating hyperplane [25]. In this machine learning technique, we basically find a hyperplane that can separate the data into two classes. This hyperplane should be able to linearly separate the patterns. Also, it can be extended to patterns that are not linearly separable by transforming data into new space. A basic working of SVM is shown in Figure 4.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

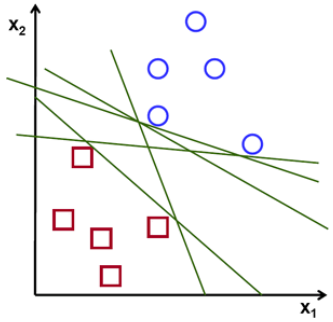

Figure 4: Basic working of SVM.

Some unique features of support vector machine are:

- These algorithms give theoretical guarantees about their performance.
- Have a modular design that allows one to separately implement and design their components.
- Not affected by local minima.
- Do not suffer from curse of dimensionality.
- They can be used for both: regression and classification tasks.

VII. ONE-CLASS CLASSIFICATION

These techniques are generally useful when there are "two-class learning" problems referred to as "target" class and "outlier" class. The target class is well-sampled whereas; the outlier class is under-sampled [26-28]. The goal of one-class classification is to distinguish between target objects and outliers by constructing a decision surface around all the target points. The end result of this would be all the normal packets (target data) will be surrounded by a hyperplane. Any incoming packet with a payload which is of different pattern will fall outside this boundary. Thus, we can train the machine with all kinds of normal traffic and any unknown attacks will automatically be filtered out. Usually there is a "rejection rule" that is taken into consideration during the training. The rejection rate makes sure that a certain percentage of training pattern lies outside the constructed decision surface. This will in turn help us to obtain a more precise description of the target class since it considers the presence of noise (unlabeled outliers) while training. In cases where the training datasets contains only pure target patterns, this rejection rate can be thought of as tolerable false positive rate [29,30].

VIII. ONE-CLASS SVM

Traditionally SVM is used for two-class classification. As described earlier, given a dataset that is not linearly separable, the SVM can separate such data using hyperplanes and kernel functions. This means that the data points that cannot be separated by a straight line in the original space I are lifted to a feature space F, where there can be a straight hyperplane that separated the data points accordingly. On projecting this hyperplane back to the original space I, this hyperplane would take the form of non-linear curve [31].

Coming to one-class SVM, this technique is a modified version of SVM, wherein only one class of data is required for classification. We will be using the one-class SVM approach that was proposed by Scholkopf et al. [32]. They mapped the data into the feature space corresponding to the kernel and to separate them from the origin with maximum margin [24]. So basically, this method separates all the data points in feature space F from the origin and maximizes the distance of this hyperplane to the origin for obtaining better margins. The result is a binary function, which returns +1 for all the data points lying in a "small region" (which represents the target data) and -1 elsewhere. To separate the data set from the origin, the following quadratic minimization function is used:

$$\omega \in F, \xi \in \mathbf{R}^{\ell}, \rho \in \mathbf{R} \ \min \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu\ell}\sum_i \xi_i - \rho$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

Subject to $(\omega.\Phi(x_i)) \geq \rho - \xi_i,\ \xi_i \geq 0$

Here, the slack variables ξ_i are introduced to allow some data points to lie inside the margin in order to avoid over-fitting. The ν determines the trade-off between maximizing the margin and the number of points lying in that margin. Thus, it sets an upper bound on the fraction of outliers and also a lower bound on the training examples used.
Solving the above equation gives us the following decision function:

$$f(x) = \text{sgn}((\omega.\Phi(x_i) - \rho) = \text{sgn}(\sum_{i=1}^{n} \alpha_i K(x, x_i) - \rho)$$

From this function, it is evident that any data x, whose value is positive, lies in the training set, or in the target region.
A popular choice for kernel function is the Gaussian kernel, which is given by:

$$K(x, x') = \exp(-\frac{\|x - x'\|^2}{2\sigma^2})$$

Where, $\sigma \in R$ is a kernel parameter and $// x\text{-}x' //$ is the dissimilarity measure.

IX. PROJECT IMPLEMENTATION

The implementation of this project has been initiated by dividing the architecture into two halves. The first half contains the intrusion detection system which is placed in-line to the network. The main function of this intrusion detection system is to analyze the packets traveling through the network and check them against rules to flag unsafe packets or intrusion attempts. An attempt to an intrusion or any packet flagged by the rule-set will be stored in a database as a report. The second half of the system is the intrusion prevention system. The function of this system is to look for unknown attacks or zero-day attacks. To do so, this system extracts suspicious packets using a sniffing tool. This packet data is then normalized for the machine learning algorithm. The normalized data is then compared to a trained model which separates the unsafe packets. For the purpose of this classification, we use a one-class classifier that is, One-Class Support Vector Machine (OCSVM). We train the OCSVM w.r.t the safe packets.
Thus, any packet that is not having similar features will get flagged as unsafe packet. Once a packet is flagged in the machine learning algorithm as unsafe, the packet details are then used by a program which updates the rules of the detection system to block the attack appropriately. It allows a node with over used battery to refuse to route the traffic in order to prolong the network life. In [6] Authors had modified the route table of AODV adding power factor field. Only active nodes can take part in rout selection and remaining nodes can be idle.

9.1 Tools Used

a) Snort:
Developed by Source fire, Snort is a free and open source network intrusion prevention system (NIPS) and network intrusion detection system (NIDS), which was created by Martin Roesch [33]. For the implementation of this system, Snort is configured as an Intrusion Prevention System. Snort uses a rule base to filter packets, hence once a packet has been identified as malicious by the machine learning algorithm, the rule base is updated via a C script and Snort can prevent those packets in the future.

b) Barnyard2 and MySQL:
Barnyard2 is an open source interpreter for Snort unified2 binary output files. Its primary use is allowing Snort to write to disk in an efficient manner and leaving the task of parsing binary data into various formats to a separate process that will not cause Snort to miss network traffic. Barnyard supports MySQL. All the reports generated by Barnyard are stored in MySQL database. This module is used for generating alerts and reports for the network administrator.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

c) Wireshark:

Wireshark is a free and open source packet analyser. It is used for network troubleshooting, analysis, software and communications protocol development, and education [34]. This system uses Wireshark as a packet sniffer. A PCAP file is generated from Wireshark which contains all the sniffed packets. This file is further sent to the KDD Extractor which further extracts all the features from packets with respect to the KDD Dataset features. The extractor generates a CSV file which is sent to R.

d) R:

R is a programming language and software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. It is extensively used by data miners for developing statistical software and data analysis. It is an important tool for computational statistics, visualization and data science. Moreover, R is easily extensible through functions and extensions, and the R community is noted for its active contributions in terms of packages. The system uses R for classifying between safe and unsafe packets. The machine learning algorithm, One-Class Support Vector Machine has been implemented through R with the help of the package: e1071. This package provides the usage of OCSVM with various kernels such as Linear, Gaussian, Polynomial and so all. We chose the Gaussian Kernel, as empirical results have shown it to be the best kernel for implementing the OCSVM. The packets are then classified and the unsafe packets are flagged and sent to the C Script for rule updating.

e) GCC:

The GNU Compiler Collection (GCC) is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU tool chain and the standard compiler for most Unix-like Operating Systems. The Free Software Foundation distributes GCC under the GNU Public License [35]. A C script was written to update the rules of Snort per the false packets detected by OCSVM.

f) KDD extractor:

The KDD Extractor is a module that takes live traffic packets or a PCAP file as input and extracts KDD features from all packets. The output is stored in a CSV file.

9.2 Training

For training the machine learning model two datasets were used. Since we are using a One-Class classification algorithm, we trained our model only for the normal or 'safe' packets. Initially we used our home network for generating a database, but due to limitations in the variety of traffic being generated we chose to use the NSL-KDD dataset along with our self-generated dataset. Another reason for using the NSK-KDD dataset was that, the system is focused on detecting and preventing probing attacks only. So, this dataset was the appropriate choice because it the data focuses on Probing Attacks such as Denial of Service (DoS) attacks, User to Root (U2R) attacks and Remote to Local (R2L) attacks. Hence, we obtained labeled data focusing on probing attacks which we used for testing purposes.

9.3 NSL-KDD Dataset

The NSL KDD Dataset is a subset of the original KDD Dataset which solves some of the inherent problems present in the original dataset. The NSL KDD Dataset does not include the redundant data present in the original dataset, thus the classifiers are not inclined towards frequently occurring entries while learning. Another big advantage is that the number of testing and training data is reasonable as compared to the original dataset. This factor makes it execution of the learning algorithms very smooth and affordable [32].

X. RESULTS

10.1 Scalability

The scalability of the system was derived using the size of the dataset and latency of the dataset as parameters. As seen in the graph (number), the size of the dataset is growing exponentially with marginal increase in latency.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

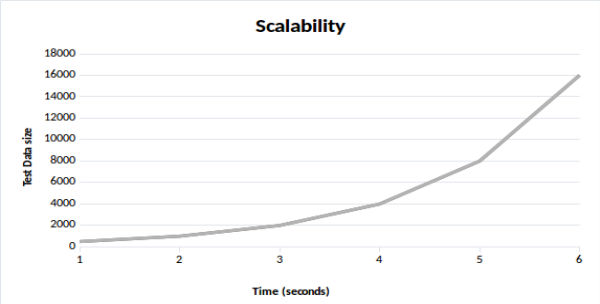

Figure 5: Scalability graph.

From the graph given in Figure 5, an exponential relation can be deduced given as:
size_of_dataset α latency
Thus, equation can be given as:

$$y = ka^x$$

Where:
y is the size of dataset
x is the latency in milli-seconds
k and a are non-zero constants which are calculated to be:

k= 0.01384
a= 9.33

Thus,

$$y = 0.01384 \times 101.3^x$$

10.2 Latency

The time required for a packet to be filtered, analyzed by the machine learning algorithm and make a rule update, if any, was calculated as latency. The datasets considered contained all unsafe packets and thus the results below are worst-case times (Table 1).

Dataset Size	500	1000	2000	4000	8000	16000
Latency(s)	3.6	4.2	3.9	5.9	4.4	5.5

Table 1: Latency observations.

As recorded, the latency increases marginally even when the size of the dataset increases more than 8 times the original. This is shown in the graph given in Figure 6.

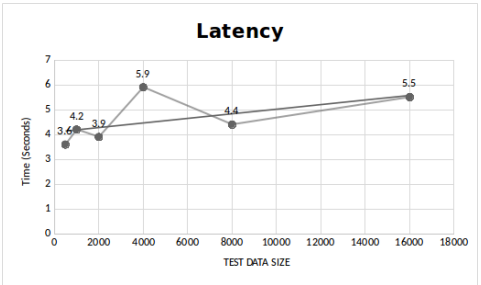

Figure 6: Latency trend.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

10.3 Accuracy

The reliability test contained calculating the percentage of false positives and negatives to determine the accuracy or reliability of the system.

Case 1:

In this case, all packets provided as input were safe packets. The number of packets input was 5000, out of which 1073 were tagged suspicious by the sniffing tool and used for analysis (Table 2).

Packets tagged as:

Accuracy: 79.87%

True	False
857	216

Table 2: Accuracy observations.

Case 2:

In this case, all packets provided as input were safe packets. The number of packets input was 10000, out of which 4872 were tagged suspicious by the sniffing tool and used for analysis (Table 3).

Packets tagged as:

Accuracy: 82.16%

True	False
4003	869

Table 3: Accuracy observations.

Case 3:

In this case, the packets provided were a mixture of safe and unsafe packets. The number of packets input was 29574, out of which 16741 were safe and 12833 were unsafe (Table 4).

Packets tagged as:

Accuracy: 74.26%

Precision: 66.15%

Recall: 82.43%

	Positive	Negative
Positive	8005	8736
Negative	4097	8736

Table 4: Accuracy observations.

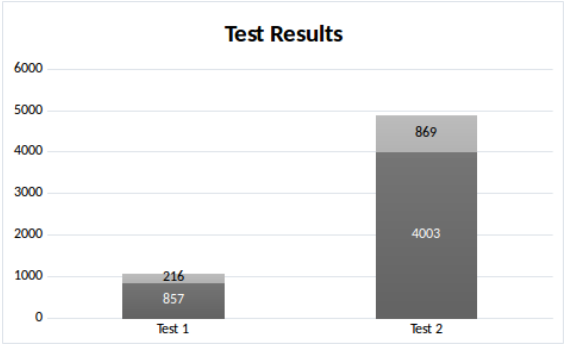

Figure 7: Test results.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

As recorded, the accuracy of the system averages at 78.76% (Figure 7).

XI. ADVANTAGES

In our proposed architecture, the IPS either blocks the packet or allows it to go inside the internal network. The packets which are blocked are sent to the IDS for further classification. As the IDS are performing one class classification the IPS will keep on performing its normal operation which is to block or to accept packets. Thus, the blocked packets by the IPS will not compromise the throughput of the system. This will help in reducing latency which is an important factor while designing an IPDS. Another advantage of this system is it reduces the number of false positives that are generated and thus gives an efficient evaluation of the threats to the network.

XII. DISADVANTAGES

In anomaly based detection it is difficult to train the system in a highly dynamic environment, since there is tremendous amount of variations in the incoming traffic making it difficult to detect an attack. Another problem is these attacks can be modeled in such a way that they are not detected as malicious by the IPS or the IDS. Once the unknown attack is detected we need to update the rule base, but the time required to update the rule base is more than the rate of incoming packets which can cause the system to slow down and increase its latency.

XIII. FUTURE SCOPE

Our system has enormous scope for development. Some of the areas it can be developed are:

a) **Range of attacks:** Our system is built only for port scan attacks; however, the machine learning algorithm can be used for any kind of network attack. This gives the scope of developing an intrusion prevent and detection system that monitors for a complete range of attacks.
b) **Accuracy:** Our system is reasonably reliable currently. However, further research into the data and developments in normalizing data would help create better trained models and improve accuracy of the machine.
c) **Latency:** Latency of the system can be improved by using faster processors or dedicated devices.

XIV. CONCLUSION

Through this project, we aim to present a theoretical and practical model for an Intrusion Prevention and Detection System using machine learning to detect and prevent known attacks as well as zero-day attacks. Using one-class classifiers, we have presented an anomaly detector that has high detection accuracy for port scan attacks at low latency rates. We have also achieved a reasonable Bayesian detection rate P (Intrusion|Alarm), i.e., the probability of having an intrusion attempt given that the Intrusion Prevention and Detection System (IPDS) raised an alarm. To conclude, the system we have developed has enormous advantages such as low latency and high accuracy as per the intention. Since the system is completely automated, it gives the opportunity for organizations to save funds as the need for a network administrator is eliminated.

XV. ACKNOWLEDGMENT

.

We owe our gratitude and respect to every person who in smallest way has taken efforts for successful completion of the seminar. Firstly, we would like to thank Dr. PN Mahalle, our seminar guide, for not only helping us choose the topic precisely but also for guiding us at every step on the way. He has invested a lot of time and efforts and has always encouraged us to move forward. We are also grateful to him for guiding and motivating us whenever in doubt. He always prompted us to give our best. We owe all our success to him. At last, we wish to thank all our colleagues and friends for providing help and support whenever needed.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

XVI. REFERENCES

1. L Ying-Dar. Creditability-Based Weighted Voting For Reducing False Positives and Negatives In Intrusion Detection. Computers and Security 2013; 39: 460-474.
2. W Su-Yun, Y Ester Yen, Data Mining-Based Intrusion Detectors. Expert Systems with Applications 2009; 36: 5605-5612.
3. L Al-Yaseen Wathiq, O Ali Zulaiha, et al. Hybrid Modified k-Means with C4.5 for Intrusion Detection Systems in Multiagent Systems. The Scientific World Journal 2015: 1-14.
4. AB Nor, S Hasimi, Identifying False Alarm For Network Intrusion Detection System Using Hybrid Data Mining And Decision Tree. Journal of Computer Science 2008; 21: 2011-2023.
5. NA Bakar, B Belaton, et al. False Positives Reduction via Intrusion Alert Quality Framework. IEEE International Conference on Communication 2005: 120-132.
6. O Al- Mamory Safaa, Z Li Hong, A Survey On IDS Alerts Processing Techniques. International Conference on Information Security and Privacy 2007: 223-253.
7. P Tadeusz, Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. Recent Advances in Intrusion Detection 2004: 102- 124.
8. M Benjamin, M Ludovic, et al. M2D2: A Formal Data Model for IDS Alert Correlation. Recent advances in intrusion detection 2002: 115-137.
9. H Emmanuel, An Intelligent Intrusion Detection And Response System Using Network Quarantine Channels: Adaptive Policies And Alert Filters. IEEE Web Intelligence and Intelligent Agent Technology 2006: 16-21.
10. H Kai, C Min, et al. Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes. IEEE Transactions on Dependable and Secure Computing 2007; 4: 190- 202.
11. K Byoungkoo, Y Seungyong, et al. ATPS- Adaptive Threat Prevention System for High-Performance Intrusion Detection and Response. Asia-Pacific Network Operations and Management Symposium 2007: 345-353.
12. X Konstantinos, GA Kostas, et al. Design and Implementation of a High performance Network Intrusion Prevention System. Institute of Computer Science, Foundation for Research and Technology Hellas 2010: 360-374.
13. GM Jose, P Vern, et al. Shunting: A hardware/Software Architecture for Flexible, High-Performance Network Intrusion Prevention. Computer and communications security 2007: 120-135.
14. Y Weinsberg, D Tzur Shimrit, et al. High Performance String Matching Algorithm For A Network Intrusion Prevention System (NIPS). IEEE High Performance Switching and Routing 2006: 102-111.
15. AB Ayesha, A Muhammad Qasim, et al. POSTER: Revisiting anomaly detection system design philosophy. Computer and communications security 2013; 1473-1476.
16. SS Nicholas, K Konstantinos, et al. On intrusion detection in opportunistic networks. International Journal of Innovation and Regional Development 2013; 67-74.
17. B Atul, T Uttam, et al. Intrusion detection in enterprise systems by combining and clustering diverse monitor data. Science of Security 2016; 7-16.
18. MM Tarek, AA Abdelmgeid, et al. A Hybrid Snort-Negative Selection Network Intrusion Detection Technique. International Journal of Computer Applications 2016; 146: 24-31.
19. X Yang, J Junchen, et al. TFA: A Tunable Finite Automaton for Pattern Matching in Network Intrusion Detection Systems. IEEE Journal on Selected Areas in Communications 2014; 32: 1810-1821.
20. P Roberto, A Davide, et al. McPAD: A Multiple Classifier System for Accurate Payload-based Anomaly Detection. Journal of Computer and Telecommunications Networking 2009; 53: 864-881.
21. V Jyothsna, VV Rama Prasad, et al. A Review of Anomaly based Intrusion Detection Systems. International Journal of Computer Applications 2011; 28: 26-35.
22. D Carli Lorenzo, S Robin, et al. Beyond Pattern Matching. Computer and Communications Security 2014; 1378-1390.
23. G Sannasi, K Kanagasabai, et al. Intelligent feature selection and classification techniques for intrusion detection in networks: a survey. Journal on Wireless Communications and Networking 2013; 271.
24. S Pham, Truong, N Uy Quang, et al. Generating artificial attack data for intrusion detection using machine learning. Information and Communication Technology 2014; 286-291.
25. P Rami, T Meytal, et al. A Decision Support System for Placement of Intrusion Detection and Prevention Devices in Large-Scale Networks. Modeling and Computer Simulation 2011; 22.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 7, July 2017

26. Md Sazzadul Hoque, Md Abdul Mukit, et al. An Implementation of Intrusion Detection System Using Genetic Algorithm. International Journal of Network Security and Its Applications 2012; 4: 109-120.
27. Z Junjie, L Xiapu, et al. Boosting the scalability of botnet detection using adaptive traffic sampling. Symposium on Information, Computer and Communications Security 2011; 124-134.
28. DV SS Subrahmanyam, A Review of Anomaly Detection Techniques in Network Intrusion Detection System. International Journal of Innovative Research in Computer and Communication Engineering 2014; 2: 6405-6409.
29. SH Rizvi RafatRana, RK Ranjit, A Review on Intrusion Detection System. International Journal of Advance Research in Computer Science and Management Studies 2015; 3: 22-28.
30. S Bernhard, PC John, et al. Estimating the Support of a High-Dimensional Distribution. Neural Computation 2001; 13: 1443-1471.
31. http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/
32. http://www.unb.ca/cic/research/datasets/nsl.html
33. https://en.wikipedia.org/wiki/Snort_(software)
34. https://en.wikipedia.org/wiki/Wireshark
35. https://en.wikipedia.org/wiki/GNU_Compiler_Collection