International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 3 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT2062111

Intrusion Detection System using Machine Learning

Jayesh Zala, Aditya Panchal, Advait Thakkar, Bhagirath Prajapati, Priyanka Puvar
Computer Engineering Department, A. D. Patel Institute of Technology, Karamsad, Gujarat, India

ABSTRACT

Intrusion Detection System (IDS) is a tool, or software application, that monitors network or system activity and detects malicious activity occurring. The protected evolution of the network must incorporate new threats and related approaches to avoid these threats. The key role of the IDS is to secure resources against the attacks. Several approaches, methods and algorithms of the intrusion detection help to detect a plethora of attacks. The main objective of this paper is to provide a complete system to detect intruding attacks using the Machine Learning technique which identifies the unknown attacks using the past information gained from the known attacks. The paper explains preprocessing techniques, model comparisons for training as well as testing, and evaluation technique.

Keywords : Intrusion Detection System, Host, Network, Detection Techniques, Support vector machine, Machine Learning, NIDS, HIDS.

I. INTRODUCTION

The Network Intrusion Detection System (NIDS) are software programs, or hardware systems that monitor and evaluate activities in a computer network for the purpose of detecting malicious behaviour and the Firewall is designed to monitor & filter the network traffic that is coming into or going out of the network. Since the frequency of attacks in the network has significantly risen, the intrusion detection program has been a required enhancement to most enterprises' protection infrastructure.

With the growing edge of technology, the use of the internet is increasingly used and with that, the security of the internet is also a concern for organizations around the world. To deter intruders from securing credential records. To protect the data Web Firewall, the network infrastructure and Internet communications are protected by encrypting, authenticating and virtual private networks (VPN). Intrusion identification is a fairly recent addition to a number of security technologies.

An intrusion requires any unauthorized access to or malicious use of the information services. A real-object is an attacker or an intruder trying to find ways to gain unauthorized access to information, harm or engage in other malicious activities. IDS is a technology that enhances network protection and protects the organization's records. The IDS assists the network administrator to track unauthorized network behaviour and alerts the administrator to the data being protected by taking necessary action. Intrusion Detection System (IDS) is a protection system that tracks and analyzes the network traffic and computer networks for possible hostile attacks from outside the organization, and for system failure or attacks within the organization.

Intrusion mitigation allows businesses to secure their networks from attacks arising from increasing

network access and information system dependency. According to the extent and complexity of current network protection risks, it would not be appropriate to discuss the usage of intrusion prevention, but the capabilities and skills of intrusion detection. Intrusion arises from: network attackers, approved users of systems who seek to achieve additional privileges for which they are not allowed; approved users who exploit privileges they have been granted.

For the recognition and prevention of attack, Network Intrusion Detection Systems (NIDS) use a Network or Host-based approach. In each case, these products check for signatures (specific patterns) showing typically malicious or suspicious intent. A variety of algorithms to classify various kinds of network intrusions have been established, but the accuracy of its findings is not verified by heuristic methods. If a succinct output metric is available, an exact efficacy of an intrusion detection program may be recorded to identify malicious sources.

II. NEED OF AN INTRUSION DETECTION SYSTEM

Now an internet day has become part of our everyday life, the world of business is linked to the internet. Each day, many people link to the Internet to benefit from the modern business concept known as e-business. Therefore, enhancing connectivity has become very relevant in today's e-business.

Across the Internet, there are two business stages. The first point is that the Internet provides an excellent business opportunity to attract customers, while still posing major risks to the company. On the internet, there are harmless and damaging users. Around the same time, a company makes its data system open to innocuous internet users. Malicious users or hackers can often have access to the internal infrastructure of the company for many reasons. There are software bugs, failure in administrative security and possibly leaving the system to the default configuration. Intruders use a range of methods, including Password cracking, peer-to-peer attack, Sniffing Attack, Eavesdropping, Application Layer Attack and more to fix the aforementioned device vulnerabilities and compromise sensitive systems. Therefore, the organization's private resources from the Internet and users inside the organization needed to be some form of protection.

Some of the common attacks against Intrusion detection system can be used are:

2.1 Denial of Service (DoS): is an attack in which an adversary directed a deluge of traffic requests to a system in order to make the computing or memory resource too busy or too full to handle legitimate requests and in the process, denies legitimate users access to a machine.

2.2 Probing Attack (Probe): probing network of computers to gather information to be used to compromise its security controls.

2.3 User to Root Attack (U2R): a class of exploit in which the adversary starts out with access to a normal user account on the system (gained either by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

2.4 Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

III. TYPES OF INTRUSION DETECTION SYSTEMS

Intrusion detection systems of two types are available. These are intrusion detection systems based on the network and intrusion detection system based on the host.

3.1 A Network IDS (NIDS): A Network IDS present in a computer or system linked to a network segment of an entity tracks and checks for ongoing attacks on that

segment of the network. Many different hacking algorithms like MD5 are used to preserve file protection in the network. If the network-based IDS is designed to learn an attack, it responds by sending administrators notices. NIDS searches for network traffic attack patterns such as large collections of related items of some kind that could mean a denial of service attack is continuing or searches for a pattern to substitute a sequence of related packets that might indicate a port scan being performed. It can be used as a monitor of the host computer on any part of a network or can be used to control all traffic between the systems which make up the whole network from a specific point in the network (the router is one example). Nids are used from which you can see traffic going into and out of a particular network segment.

3.2 Host-based Intrusion Detection System (HIDS): is installed on a single device or server, known as a host, and tracks the operation of the host only. Host-based intrusion detection technologies can also be divided into two categories: signature-based (i.e. misuse detection) detection and anomaly-based technology. HIDS tracks key system file status and detects the creation, alteration and deletion of the monitored files by an intruder. The HIDS then triggers a warning to change a file attribute, to build new files, or to remove existing file.

IV. METHODS OF INTRUSION DETECTION SYSTEMS

4.1 SIGNATURE BASED INTRUSION DETECTION SYSTEM: Signature-based IDS refers to attack detection by looking for specific patterns such as network byte sequences and malicious malware-based sequences. Anti-virus software produces the terminology which calls these detected patterns signatures. While IDS based on a signature can easily detect known attacks, new attacks can not be detected for which there is no pattern. This method has the signature to identify the intruder automatically. Misuse detection is automatically generated and the work is more complex and precise than it is performed manually. It should be submitted to the right authorities in compliance with the robustness and seriousness of a signature that is triggered in the network.

4.2 ANOMALY BASED INTRUSION DETECTION SYSTEM: An intrusion detection system for anomalies provides a mechanism for the identification and maltreatment of both network and computer intrusions through the monitoring and classification of device behaviour as normal or anomalous. The classification consists not of patterns or signatures, but of certain rules and seeks to detect malicious activities of some kind that are not natural to operate the program. In the case of signature-based systems, attacks that had previously been produced can be detected only. The advantages of this system are the Possibility of identifying new attacks as intrusion; inability to identify the triggers and properties of new attacks; decreased reliance on IDS (as opposed to attack-based signature-based); and the ability to detect misuse of user rights.

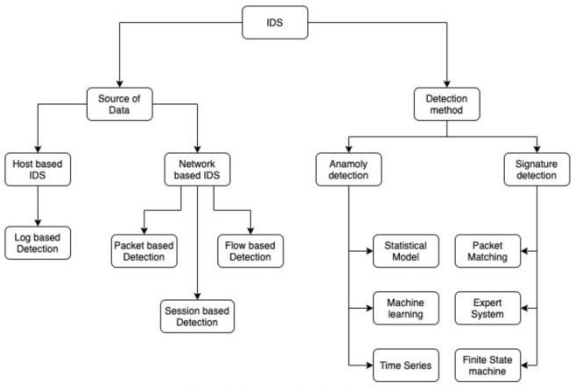

Figure 1: Types and methods of IDS

V. WORKFLOW

The data that comes over the Internet is in the form of a stream of data packets, packets are units of data made into a single package that travels along a given network path, it navigates over the web and transmits the data. The pattern in which they transit or its types

helps us to detect various types of attacks. On proper monitoring of network traffic in real-time one can check if there are any dangerous payloads.

These packets transit over the internet, so our NIDS will be at the perimeter, separating your internal network from the outside world. Placing a NIDS like before, after, or inside the firewall in any network can be changed as per requirements. If one wants NIDS to see all the traffic, then one should follow the scheme: Internet > Router > NIDS > Firewall > Switch > Internal network. If, on the other hand, one wants the NIDS placed before the firewall, then it will only be able to see the traffic passing through the firewall. In that case, scheme: Internet > Router > Firewall > Sensor > Switch > Internal network will be followed. One can also place the NIDS anywhere to monitor traffic, such as on the network where critical servers are hosted.

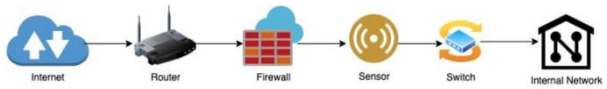

Figure 2: Incoming packet traversal

5.1 Packet Collection: The data is passed to this module as an IDS entry. The data will be saved and processed in a server. IDS collects data packets and changes them on the network-based system and collects information in host-based IDS such as disk utilization and device processes. It is then passed to the Packet Decoder.

Let us consider a set C = {c_1, c_2, c_3, …, c_n} where c_1, c_2, c_3, ..., c_n are the network packets passed to the Packet Decoder.

5.2 Packet Decoder: A PPP connection, an Ethernet (copper or fibre-optic) connection, or whatever. The purpose of the Packet Decoder is to prepare the packets for the Preprocessors by decoding the information like source, destination, length, protocol, method, status code etc.

After decoding we obtained a new set P = {p_1, p_2, p_3, …, p_n} where p_1, p_2, p_3, …, p_n are the decoded network packets.

5.3 Analysis and Classification:

5.3.1 Dataset: Technology is changing quickly day by day, and many inventions and technological advances are made to protect computer systems against any attacks by network intrusions. The available KDD Cup 1999 data set was usually used for network intrusion research. This data set is applied to different algorithms for machine learning. But there were many problems with this dataset. First of all, the number of redundant records in the dataset is a major disadvantage of the KDD dataset. 78% of records have been duplicated in the training set, and about 75% of the total number of records has been duplicated in the testing dataset and our results are thus translated to biased learning algorithms. Though there can also be a new version of the KDD Cup 99 dataset, the NSL KDD dataset, which now uses the dataset for use in machine learning algorithms, is available as a public data set for the network intrusion detection system. The new NSL KDD Test and Train dataset combine only highly selected data from the original KDD dataset and there is no redundancy. Consequently, the results of the research evaluation are declared as a standard dataset and consistent in every research. In principle, the training dataset consists of four types of attacks. They are first the Denial of Service(DoS), second the Probe Attack, third is called the User to Root(U2R) and last the Root to Local(R2L) root. This, in turn, is composed of over 21 attacks.

Let D = {d1, d2, d3, …, d_i} be the dataset which contains m number of packets where

each d1, d2, d3, ..., d_i are configured by 41 attributes(features).

5.3.2 Preprocessing: The Pre-processing of data is the conversion of raw data formats to usable and efficient formats which will be passed over to the models for training and testing. The primary phase of pre-processing performed here is mapping of the data entries in the data set with the segregated attack classes which are DoS, Probe, U2R and R2L and are accomplished by the lambda function.

Therefore, set D is further divided into subsets $M1 = \{m1_1, m1_2, m1_3, \ldots, m1_w\}$, $M2 = \{m2_1, m2_2, m2_3, \ldots, m2_x\}$, $M3 = \{m3_1, m3_2, m3_3, \ldots, m3_y\}$ and $M4 = \{m4_1, m4_2, m4_3, \ldots, m4_z\}$ such that $w + x + y + z = i$.

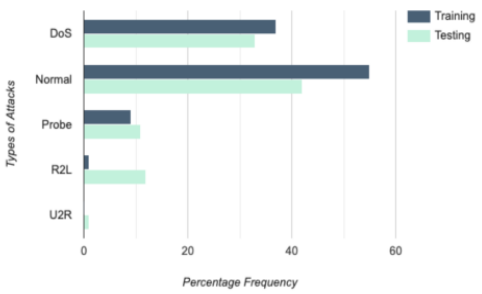

Figure 3: Distribution of Attack class

Here the normal class does not represent any type of attack on the intrusion system for network detection and it shows the intrusion detection system's ordinary condition.

5.3.3 Feature Selection: First, it is very difficult to select the feature from the dataset, as this will tell us how important a feature is. With the change in the attack type the feature selection changes. Secondly, there is no real-time networking labeled traffic. Any redundant data can be eliminated by selecting the entire problem from the sub-set of features. Once done, we fit the test and training dataset's data framework. We create two target classes, one is known as normal and the other is known as an attack. There is a function that divides the data set into two class attack labels.

There are 41 features listed in the NSL KDD dataset. Since it is not possible to take into account all 41 features, we have drawn a key graph. The characteristic graph gives us a clear picture of all characteristics and their variation. This is why we decided to drop the last 9 because the value of the latter is not changing. Some of the columns like srv_rerror_rate, urgent,is_host_login, su_attempted, num_shells, and land have only 0 as their value. We, therefore, decided to put them out of the dataset preprocessing. Then, by changing the number of attributes from float16 and Int16 and then adding this to a new numerical data frame, we extract and scale it to have zero mean and the unit variance.

Whilst applying feature reduction on the set M1 each element of M1 i.e. $\{m1_1, m1_2, m1_3, \ldots, m1_w\}$ consists of 41 attributes and 9 of them will be dropped due to lesser importance thereby, the new set obtained will be $M1' = \{m1_1', m1_2', m1_3', \ldots, m1_w'.\}$ Similarly, M2, M3 and M4 will also be reflected with the same changes respectively.

Another major part of the data package preprocessing includes how the categorical attributes are dealt with. First, we extract the attributes of both training and test data sets. When done, the LabelEncoder from the Scikit Learn package encodes categorical attributes.

One Hot Encoder is a mechanism that transforms the categorical variable, accepting value 1 if true, and value 0 when false. This type of encoding is very easy to use, but you can try to do it with scicitlearn in Python because it does not currently have a simple method of transformation.

The pre-processing operation has to be performed because both the numerical and non-numerical records are included in the dataset. The scikit works perfectly with numerical values, so we try to convert the records in categorical form into binary. All the features of each and every packet will undergo transformation to obtain binary values.

On applying the binary function on dataset M1' = {$m1_1$', $m1_2$', $m1_3$', …, $m1_w$'} a new set B1 is obtained; f(M1') = B1 where elements of B1 = {$b1_1$, $b1_2$, $b1_3$, …, $b1_w$} are the binary values of novel set B1. Likewise B2, B3, B4 are also obtained from M2', M3', M4'.

The features are scaled in a way where large value features that can exceed the result are avoided.

5.3.4 Data Sampling: This technique uses randomization to ensure that every element of the population has equal opportunities to participate in the selected sample. Alternatively, it is referred to as random sampling. Since the values of the original data sample will contain different values for each of the four attacks as well as normal, we attain are sampled data set where the number of elements for each data set are equal.

On completion of data sampling, these sets B1, B2, B3 and B4 will have the same number of packets x, where x is maximum of either w, x, y or z.

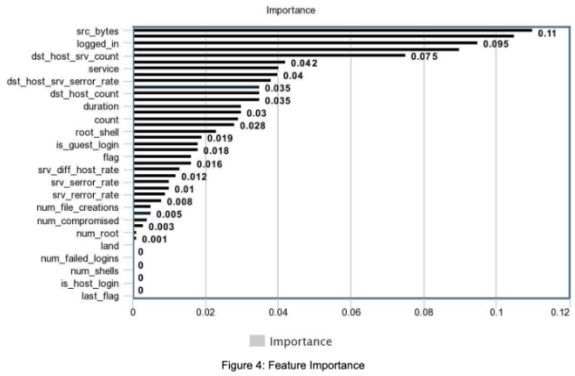

Figure 4: Feature Importance

5.4 Training and Testing Models:

Training data is used to ensure that the machine recognizes patterns in the data, the cross-validation data is used to ensure the algorithm used to train the machine is more accurate and effective, and the test data is used to see how well the machine can predict new answers based on its training. Comparing few models hereby,

5.4.1 Support Vector Machine: In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Based on a collection of training examples, each of which is classified as belonging to one or the other of two categories, an SVM training algorithm generates a model which assigns new examples to one or the other category, rendering it a non-probabilistic binary linear classifier (although methods such as Platt scaling exist to use SVM in probabilistic classification).

Results of SVM:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.98719	0.83662	[5272 2186] [619 9092]	
Classification report			
Precision	**Recall**	**f1-score**	**Support**
0.89	0.71	0.79	7458
0.81	0.94	0.87	9711
Average			**Total**
0.84	0.84	0.83	17169

5.4.2 Naive Bayes Classifier: In machine learning, naïve Bayes classifiers are a family of basic "probabilistic classifiers" focused on the interpretation of Bayes 'theorem with clear (naïve) assumptions of independence among the features. They are among the simplest configurations of Bayesian network models. Naïve Bayes has been widely researched since the 1960's. It was implemented in the text-retrieval group in the early 1960s, and remains a popular method for text categorization, the problem of judging documents as belonging to one category or the other with word frequencies as the features. In this domain, it is competitive with more advanced methods with adequate preprocessing, including support vector machine. It is also used for automatic medical diagnosis.

Results of Naive Bayes Classifier:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.87839	0.83365	[5487 1971] [885 8862]	
Classification report			
Precision	Recall	f1-score	Support
0.86	0.74	0.79	7458
0.82	0.91	0.86	9711
Average			Total
0.84	0.83	0.83	17169

5.4.3 Decision Tree: A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g., if a coin flip comes up heads or tails), each branch represents the test result, and each leaf node represents a class mark (decision made after all attributes have been computed).

Decision tree classifiers are used in various pattern identification issues such as image classification and character recognition as well as a well-known classification technique. Decision tree classifiers are more successful due to their high adaptability and computational efficiency, particularly for complex classification problems. In addition to numerous typically supervised classification methods, decision tree categories are above expectations

Results of Decision Tree:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.99720	0.81659	[5591 1867] [1282 8429]	
Classification report			
Precision	Recall	f1-score	Support
0.81	0.75	0.78	7458
0.82	0.87	0.84	9711
Average			Total
0.82	0.82	0.82	17169

5.4.4 Random Forest Classifier: Random forest, as its title suggests, is representation of a large number of individual decision trees which act as an ensemble. Every single tree in the random forest spits out a class prediction and the class with the most votes is the prediction of our model.

The reason the random model of forests works so well is: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. And uncorrelated models can generate more accurate set predictions than any of the individual predictions.

The explanation for this wonderful effect is that trees shield each other from their individual errors. Because some trees will be incorrect, several other trees would be correct and the trees will move in the right direction as a group. So the prerequisites for successful production of random forests are:

i. The features need to contain some real signal so that models developed using those features do better than random conjectures.

ii.The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

Results of Random Forest Classifier:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.99808	0.82952	[5489 1969] [958 8753]	
Classification report			
Precision	Recall	f1-score	Support
0.85	0.74	0.79	7458
0.82	0.90	0.86	9711
Average			Total
0.83	0.83	0.83	17169

5.4.5 k-nearest neighbors: *k*-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until function evaluation.

A useful strategy for both classification and regression may be to assign weights to neighboring inputs, so that the nearby neighbors contribute more to the average than the more distant ones. A typical weighting scheme, for example, is to assign each neighbor a weight of 1/d, where d is the distance from the neighbour.

Neighbors are taken from a set of objects for which the class (for classification k-NN) or the value of the object property (for regression k-NN) is defined. This can be called the training set for the algorithm, although no specific training phase is required. A peculiarity of the k-NN algorithm is that they are prone to the data's local structure.

Results of k-nearest neighbors:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.99144	0.86662	[5787 1671] [619 9092]	
Classification report			
Precision	Recall	f1-score	Support
0.90	0.78	0.83	7458
0.84	0.94	0.89	9711
Average			Total
0.87	0.87	0.86	17169

5.4.6 Logistic Regression: Logistic regression is a statistical model that uses a logistic function to model a binary dependent variable in its basic form, though there are several more complex extensions. The logistic regression (or logit regression) is estimating the parameters of a logistic model (a type of binary regression) in regression analysis.

Results of Logistic Regression:

Cross validation mean score	Model Accuracy	Confusion Matrix	
0.93954	0.84187	[5963 1495] [1220 8491]	
Classification report			
Precision	Recall	f1-score	Support
0.83	0.80	0.81	7458
0.85	0.87	0.86	9711
Average			Total
0.84	0.84	0.84	17169

Let there be a set R = {r_1, r_2, r_3, ..., r_p} where r_1, r_2 , r_3 ,… , r_p are the patterns recognised by the model where every pattern will have distinct order and number of packets.

5.5 Detection Engine: The "**Detection Engine**" works in the real world environment to analyze the pattern of packets and it's content to recognize if there is any sort of attack. If a packet ends up matching any of the rules, it notifies an appropriate alert message to the security administrator.

Set R= {r_1, r_2, r_3, ..., r_p} contains various patterns which has ordered packets which will be matched to the incoming packets P = {p_1, p_2, p_3, …, p_n} and if the sequence of the packets from set P are matched with any of the pattern from set R then that particular attack will be detected.

5.6 Alert: This explains the system's reaction and attack. This may either warn the system administrator using an e-mail/alarm icon that the system administrator is responsible for all necessary data, or it

may disable the system by discarding packets so that they do not reach or close the ports.

Alerts are logged to "**Log Packet Analysis**" which sends the information out for logging in a **lof-file** format, such as tcpdump and .pcap files. Network Administrators can monitor these log files for further inspection.

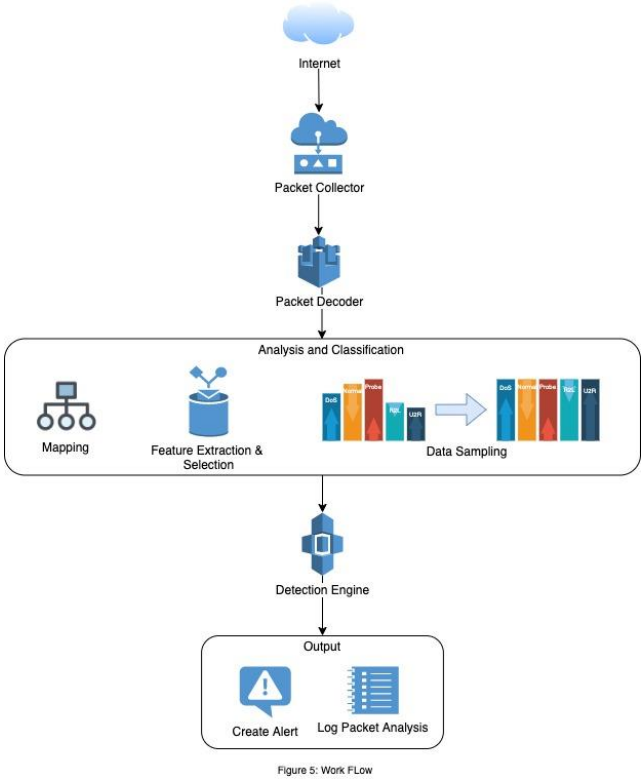

Figure 5: Work FLow

VI. Empirical Result Analysis

Comparison of various models considered for the testing are juxtaposed in the below, the table lists various attributes which have a major impact on selecting a preferred model for the domain.

Accuracy is basically the scale of predicting a correct class for a given observation. Two very important model assessment methods are precision and recall. Where accuracy refers to the proportion of your outcomes that are relevant, refers to the percentage of total relevant results correctly classified by your algorithm. Cross validation is used to assess the predictive performance of the models and to assess how they are performed in a new set of data known as test data outside the sample

Comparison Table:

	SVM	Naive Bayes	Decision Tree	Random Forest	KNN	Logistic Regression
Accuracy	0.83	0.83	0.81	0.82	0.86	0.84
Precision	0.89	0.86	0.81	0.85	0.90	0.83
Recall	0.71	0.74	0.75	0.74	0.78	0.80
Cross-Validation	0.98	0.87	0.99	0.99	0.99	0.93

A bar chart representation is depicted below where output fields are compared. According to the output we can justify that k-nearest neighbors model has the highest accuracy and is best suited to complete the task.

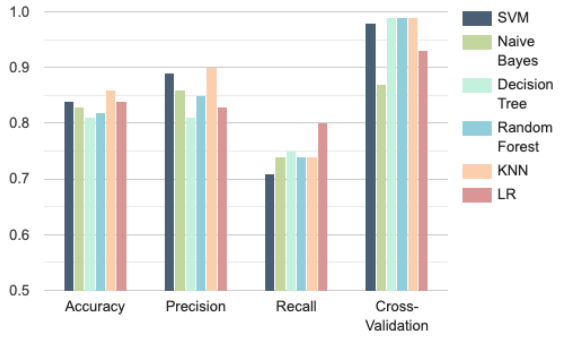

Figure 6: Comparison of models

VII. OUTPUT LOG FILE

The attached output picture contains logs of a Probing attack on a local system whilst testing.

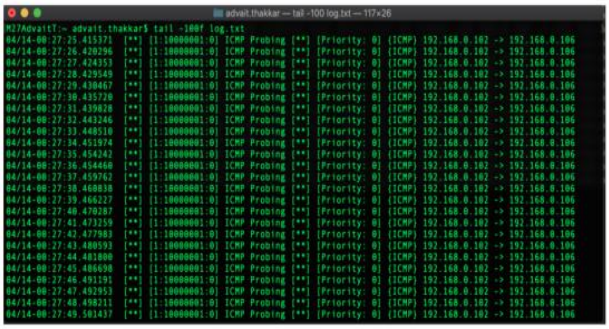

Figure 7: Screenshot Output Log

VIII. FUTURE WORK

In the future, the number of tests for our system will increase and we will find different accuracies. We hope to improve the genetic algorithm to surge IDS precision. The current system displays only log information but uses no techniques to analyze the information in the logs and to extract information. Data mining techniques can be utilized to analyze the information in log files to help efficient decision-making to enhance the system. Only the known attacks are detected by the current system. In order to gain knowledge by analyzing increasing traffic, and to learn new patterns of intrusion this can be extended by integrating intelligence into them.

IX. CONCLUSION

The IDS offers basic detection technology to safeguard network systems that are connected directly or indirectly to the Internet. But it's up to the network administrator finally at the end of the day to make sure that the network remains safe. It doesn't protect the network completely from intruders, but IDS helps the network administrator track bad people on the Internet whose very aim is to make your network infringe and vulnerable to attacks. After firewall technology is deployed in the network perimeter, IDS becomes the main part of many organizations. In case of traffic not crossing the firewall, IDS can provide protection against outside users and internal attackers.

We have proposed to develop a NIDS using Machine Learning technique that will learn from the past data and stop the intruders or will notify the network administrator. The system offers to stop a wide range of intrusion attacks and learn to stop the new and unhackneyed attacks.

IX. REFERENCES

[1]. C. Chang and C. J. Lin, LIBSVM, "A Library for Support Vector Machines", the use of LIBSVM, 2009.

[2]. Rung-Ching Chen, Kai-Fan Cheng and Chia-Fen Hsieh, "Using Rough Set and Support Vector Machine for Network Intrusion Detection", International Journal of Network Security & Its Applications (IJNSA), Vol 1, No 1, 2009.

[3]. Need and study on existing Intrusion Detection System. Available at: http://www.sans.org/resources/idfaq.

[4]. Resources about packet capturing. Available at: http://www.netsearch.org/jpcap.

[5]. Salvatore Pontarelli, Giuseppe Bianchi, Simone Teofili. Traffic-aware Design of a High Speed FPGA Network Intrusion Detection System. Digital Object Identifier 10.1109/TC.2012.105, IEEE TRANSACTIONS ON COMPUTERS.

[6]. PrzemyslawKazienko&PiotrDorosz.IntrusionDetection Systems (IDS) Part I - (network intrusions; attack symptoms; IDS tasks; and IDS architecture). www.windowsecurity.com › Articles & Tutorials

[7]. Sailesh Kumar, "Survey of Current Network Intrusion Detection Techniques", available at http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids.pdf.

[8]. Dataset: https://www.unb.ca/cic/datasets/nsl.html

[9]. AHMAD, M. BASHERI, M. J. IQBAL, and A. RAHIM, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection." Online]. Available: 0.1109/ACCESS.2018.2841987

[10]. H. Nkiama, S. Z. M. Said, and M. Saidu, "A Subset Feature Elimination Mechanism for Intrusion Detection System," (IJACSA) International Journal of Advanced Computer Science and Applications,, vol. Vol. 7, no. No. 4, 2016.

[11]. "Sparsity-driven weighted ensemble classifier." Online]. Available: https://arxiv.org/abs/1610.00270

[12]. Prof.S.S.Manivannan and Dr.E.Sathiyamoorthy, "An Efficient and Ac-curate Intrusion Detection System to detect the Network Attack Groups using the Layer wise Individual Feature

[13]. S. Revathi and D. A. Malathi, "A Detailed Analysis on NSL-KDD Dataset Using Various Machine Learning Techniques for Intrusion De-tection," International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 12, 2013.

[14]. R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying con-volutional neural network for network intrusion detection." in ICACCI 2017, pp. 1222–1228.

[15]. K. S. Desale, C. N. Kumathekar, and A. P. Chavan, "Efficient Intrusion Detection System using Stream Data Mining Classification Technique,," in International Conference on Computing Communication Control and Automation,, 2015.

[16]. Q. Niyaz, M. Alam, W. Sun, and A. Y. Javaid, "A Deep Learning Approach for Network Intrusion Detection System,," in Conference Paper in Security and Safety, 2015.

[17]. "Mrutyunjaya Panda and Manas Ranjan Patra, "Network Intrusion Detection using Naive Bayes"," International Journal of Computer

[18]. "Types of Intrusion Detection System." Online]. Available: https://en. wikipedia.org/wiki/Intrusion_detection_system

[19]. K. A. I. PENG, V. C. M. LEUNG, and Q. HUANG, "Clustering Approach Based on Mini Batch Kmeans for Intrusion Detection System Over Big Data," SPECIAL SECTION ON CYBERPHYSICAL- SOCIAL COMPUTING AND NETWORKING, , 2018. Online]. Available: 0.1109/ACCESS.2018.2810267

[20]. H. su Chae and S. H. Choi, "Feature Selection for efficient Intrusion Detection using Attribute Ratio," INTERNATIONAL JOURNAL OF COMPUTERS AND COMMUNICATIONS , vol. Volume 8, 2014.

[21]. SPECIAL SECTION ON CHALLENGES AND OPPORTUNITIES OF BIG DATA AGAINST CYBER CRIME, 2018. Online]. Available: 10.1109/ACCESS.2018.2854599

[22]. Vipin Das , Vijaya Pathak, Sattvik Sharma, Sreevathsan, MVVNS.Srikanth, Gireesh Kumar T," NETWORK INTRUSION DETECTION SYSTEM BASED ON MACHINE LEARNING ALGORITHMS" International Journal of Computer Science & Information Technology (IJCSIT), Vol 2, No 6, pp 138-150, December 2010.

[23]. Majed Tabash, Tawfiq Barhoom," An Approach for Detecting and Preventing DoS Attacks in LAN," International Journal of Computer Trends and Technology (IJCTT) – Volume 18 Number 6, pp 265-27, Dec 2014.

Cite this article as :

Jayesh Zala, Aditya Panchal, Advait Thakkar, Bhagirath Prajapati, Priyanka Puvar, "Intrusion Detection System using Machine Learning", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6, Issue 3, pp.61-71, May-June-2020. Available at doi : https://doi.org/10.32628/CSEIT2062166

Journal URL : http://ijsrcseit.com/CSEIT2062166