

Internet Monitoring using Snort and Naïve Bayes Method using Internet of Things (IoT)

Shikha Verma¹, Neha Gupta², Rosey Chauhan³ and Gunjan Srivastava⁴

¹JSS Academy of Technical Education, Noida/Information Technology, Noida, India
Email: shikhaverma@jssaten.ac.in

²⁻³JSS Academy of Technical Education, Noida/Information Technology, Noida, India
Email: {neha.gupta, roseychauhan}@jssaten.ac.in

⁴Intune Administrator, Yoda Tech Pte. Ltd., Singapore
Email: sriv.gunjan@gmail.com

Abstract—This paper describes a novel technique for identifying malicious users using deep packet-sniffing using the "Snort" programme. How to track a person's online behavior and block hostile websites that are or could encourage the use of data piracy are used for tracking the unintentional malicious acts and preventing the specific user from visiting the particular website again. In this work, the Naive Bayes algorithm was used to predict attribute- classes utilizing training data-sets from NSLKDD in the KDD99 format and testing data collected from the DoS attack recording process on IoT-based devices.

Naive Bayes has the benefit of requiring less training data, which makes it easier to get the estimated parameters required for classification. Using the findings of the research, it was possible to identify assaults on IoT devices. Snort tools are used to collect traffic records. The KDD99 format was then applied to the log's findings, and the Naive Bayes algorithm was used to analyze them. For testing purposes, data from the Raspberry Pi 3's IDS log process and a training dataset from NSLKDD in KDD99 format are both employed in this study.

Index Terms— Snort, IDS, IPS, packet filtering, Naïve Bayes.

I. INTRODUCTION

“Monitorization is tracking the user particular activities that the user is searching for, on the internet in a particular network. The user may get reported if the user is found doing something suspicious on the internet i.e., visiting the torrent sites, malicious sites. Intrusion detection is the process of observing and evaluating sequential computer system behavior for indications of security issues. The vision of intrusion detection right now is mostly on the feature selection issue. The performance of an intrusion detection system (IDS) is decreased as a result of some characteristics being redundant and unnecessary, which prolongs the detection process [1]. IoT has gained a lot of familiarity recently. A growing number of commonplace gadgets are being connected to the Internet, which offers various features including sensing, autonomy, and Contextual-awareness [2]. IoT interconnection-networks may link different things with IP addresses and recognized identities so they can talk with one another and share data [3].

It helps the society in easy configuration of the snort, cost effective over hardware firewalls and more effective than firewalls. Firewall usually applied to the network whereas the snort IDS/IPS applied to the network and host level. In the majority of complicated real-world situations, naive Bayes frequently outperforms expectations [4].

In contrast to the earlier study, the latest research recommended using testing data from the log process of DoS attacks on IoT-based devices along with training data-sets from NSL-KDD using the KDD99 format to predict the likelihood that the attribute classes would result in aberration or normal. Naive Bayes was the research methodology employed. This study employed snort-IDS on IoT as testing-data and NSL-KDD data as training data.

II. LITERATURE SURVEY

Naive Bayes produced very competitive results despite having a straightforward structure, according to research [5]. Depending on whether it deals with the full assault, divides it into four primary categories, or only concentrates on regular and anomalous behavior, there are three levels of attack granularity.

The earlier study described how IDS makes use of the KDDCUP99 data set. NSL-KDD data were used in this investigation. A training-set and a testing-set of data have been made available by NSL-KDD for the research process' assault categorization. The Naive Bayes approach will be used to classify assaults based on the NSL-KDD data in order to classify fresh attacks. The KDDCUP (Year 1999) data-set, which is preparing data based on DARPA (year 1998) data offered for intrusion detection system design with the aim of evaluating different intrusion detection methods. The Naive Bayes approach has been successfully used in research to categorize assaults with an accuracy range of 82–85.67% [6].

III. INTRUSION DETECTION SYSTEM (IDS)

An IDS is basically a system that is capable of finding questionable behavior in a network or system. If there are any unusual behaviors the IDS will alert the system or network administrator if anything connected to network traffic is discovered. An IDS can be used for a variety of purposes [7], including:

- a. Prevent the growing security danger, as many unlawful acts are discovered by careless people, and these activities are punished.
- b. Identify security attacks and vulnerabilities on network systems that are not blocked by common solutions like firewalls, as this creates many security gaps. Users and administrators make mistakes when configuring and utilizing the system, and users who don't comprehend the system also contribute to security flaws in the networks and protocols they use.
- c. Following the discovery of the initial attempt, the attacker will frequently choose a system to attack by investigating or testing the information systems that would be the target in order to locate the entry points.
- d. Keep protected files off-network.
- e. In the role of controller for security administrators and designers, particularly for quick-growing businesses.
- f. Directly provide accurate facts about the disruption, enhance diagnosis and recuperation, and address any attack-causing variables.

IV. DATASETS

NSL-KDD Dataset, for several inherent issues stated in the KDD (year 1999) dataset, NSL-KDD is a recommended dataset. The new KDD dataset has a few issues and might not be an accurate representation of the actual network because there aren't any publicly accessible data-sets for network-based IDS, but we still believe it can be used as a useful validation set to help researchers compare various intrusion detection methods [8]. Additionally, the NSL-KDD training and testing sets include comparable numbers of records. This benefit eliminates the requirement to choose a piece at random, making it feasible to conduct experiments on the entire collection. The evaluation results from multiple research studies will therefore be comparable and consistent [9].

Format for KDD'99 Datasets. In order to compare various intrusion detection approaches, the KDD (year 1999) dataset (kdd.ics.uci.edu) pre-processes data based on year 1998 DARPA data is offered for intrusion detection system design. Pre-processing of tcpdump data was done in 1999 for the "International Knowledge Discovery and Data Mining Tools Competition" activity, which involved intrusion detection [10]. The three feature groups that make up the 41 features of the KDD year 1999 format are the fundamental properties of single Network connections, the material features in the connections advised by the subdomain, and the traffic features computed using time [11].

V. METHODOLOGY

Identification of the System

The research project, "DoS Attack Detection Using the Naive Bayes Method on the Internet of Things (IoT) Technology- Based Device," is now in its design and implementation stages. It consists of a number of parts in the form of:

- a. Hardware device preparation;
- b. Simulated attacks;
- c. KDD99 format testing data preparation
- d. Creating data training using NSL-KDD data, which is available at www.unb.ca/cic/datasets/nsll.html.
- e. Using feature selection to identify the dataset's most important feature subset
- f. Using the Naive Bayes approach to categorize different attack types as either normal or anomalous
- g. needs a web interface to display the classification of attack type outcomes.

Hardware device preparation is done in step one of the processes. Routers, switches, access points, PC clients, and a Raspberry Pi 3 as an IDS are among the used gadgets. The Mikrotik RB450R2 Router is used as the router in this study, with port 1 linked to the internet and port 2 to the switch. A switch is used to link all devices, including a computer, an access point, and a raspberry pi 3.

In this paper, a technique for early intrusion detection utilizing Snort software is proposed. The Multi - layer Feed-forward Neural Network (MLFNN) using Back-Propagation (BP) method was trained using the data that was gathered. Software called MATLAB was used to simulate the MLFNN with BP method. [12] Modern computing architecture must include Network Intrusion Detection Systems (NIDS) to monitor unwanted and harmful network activity (such as unauthorized system access or poorly configured systems). The bulk of industrial NIDS are signature-based, which means that they look for certain patterns in network traffic to establish what constitutes undesired traffic. Despite the fact that such systems are quite efficient against possible attacks, signature-based identification fails when attack pathways are unidentified or when known attacks are changed to circumvent such restrictions. In real-world circumstances, signature-based identification in NIDS is usually plagued by false - positive and struggles to detect unknown or updated threats. The sensing of malicious shellcode, a high impact threat vector that allows attackers to gain unauthorized access to command lines on both traditional computer systems and cyber-physical systems like the smart grid infrastructure, is particularly problematic because shellcode patterns can be challenging to differentiate from benign network activity.

For instance, while utilizing network intrusion detection techniques while serving as a network security specialist for the Shop Direct Group (UK). It was discovered that signatures intended to match shellcode commonly also recognized other non-shellcode binaries, like as DLLs and jpg picture files, by Sguil and Snort from the Debian-based Linux distribution Security Onion. Due to the number of these false positives, it became necessary to disable the signatures themselves, rendering them unusable. Shellcode and signature-based systems frequently encounter the false positive issue, which Microsoft extensively covers in their patent for techniques to identify malicious shellcode with fewer false positives in memory [13].

A. Data-Collection Simulation Testing

I'm wrong! No such source was found. According to the specified IDS design for this simulation, the Router should have a core switch nearby, which all users must pass through in order for the IDS server to be able to keep track of their activity. No such source was found. The attacker will attempt to impede users' access to the network system in a variety of ways, such as network flooding, which includes overburdening the network with data. The system will recognize an attack when it is launched by the attacker. All data that has been filtered by the system will be recorded in a log and processed for data testing. Log data will be translated to testing using the KDD99 format.

Used Dataset: 3.4, The training dataset and testing dataset are described in this section. The NSL KDD training, which can be found at www.unb.ca/cic/datasets/nsll.html, is where the training data is taken from. This study used all of the 125,973-training data, which included 67,343 normal class data and 58,630 anomaly data.

By identifying shellcode trends in network traffic using a neural network-based artificial classifier, the intelligent system for detecting intrusions described in this study greatly outperforms signature-based detection techniques. The ANN- based classifier displays outstanding precision in addition to 100% sensitivity on the testing data (identifying all instances of shellcode). By testing on a very large collection of benign internet traffic file content (400,000 samples), the performance of the suggested technique was then further assessed with regard to the false-positive rate, where the suggested technique attained a rate of false positives of less than 2%. The use of network attack systems in the real world is very concerned with minimizing the false-positive rate since elevated amounts of false - positive lead to a very poor signal-to-noise ratio and frequently render the system unusable.

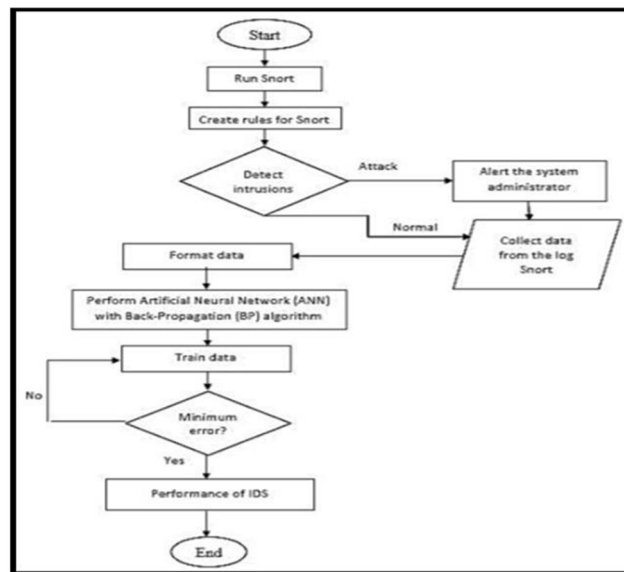


Fig 1. General snort rules and working

B. Naive Bayes Method

A straightforward probabilistic-classifier that uses the Bayes Theorem under the condition of high-impedence is the Naive Bayes Classifier (NBC). [14] DoS Attack Detection employs the Naive Bayes algorithm as part of the IDS system implementation. Naive Bayes is frequently applied in the classification of text-based documents. Continuous value attribute is still used in data today. This continuous characteristic can be handled in a variety of ways, including one where the continuous attribute data is modelled or computed straight into the probabilities. The Gaussian function is the most frequently utilized [15]. This technique has the benefit of a straightforward algorithm with low computational complexity.

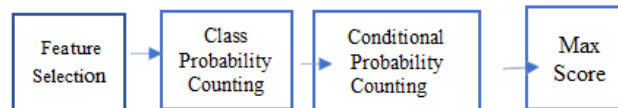


Figure 2. The flow of the Naive Bayes method

In pre-processing, feature selection is a crucial approach that is frequently applied. With the help of this technique, fewer features are needed to determine a target-class value, and superfluous or non-relevant features and data that lead to misinterpretations of the target-class are also reduced [16]. The two stages of the feature-selection approach employed in this research will be conducted utilizing Information Gain (IG) and Correlation Feature Selection (CFS). The Info Gain evaluator is used in the initial stage of the experiment to pick features utilizing strategies for filtering out irrelevant elements. Using a scoring method for minimal or continual feature weighted that is reduced using maximum entropy, IG is a feature selection strategy.

The following step is feature selection using CFS based on their correlation. The CFS approach is a component of the heuristic evaluation that takes into account both the degree of inter-correlation among the features as well as the advantages of each feature for class predictions. A subset of data with parameters that have a high correlation to a category receives high marks from the CFS, but weak connections between them. [17]. The most valuable traits, those that express their eligibility for selection, will be chosen by CFS. Nine fields were chosen as references for the training data and testing data as a consequence of the 40-field feature selection. Service, flags, src bytes, dst bytes, similar srv rate, diff srv rate, dst host srv diff host rate, and dst host srv error rate are all variables that make up this list. Class Probability Counting. Class Probabilities represent the likelihood that a certain class of problems will arise given the anticipated outcomes. Since there are two classes of problems in the dataset, the probability (P) for each class may be calculated by dividing the total value of the data frequency related class by the value of the frequency of each class of problems. As a result, the probability value for each class of problem is calculated using the available training data.

Conditional probability counting is the probability of each input value compared to the class value is then calculated as the following step. There are two different types of calculations in the calculations.

- a) Determining Mean Value.
- b) Calculating the Standard Deviation Value, Using the Gaussian Function to Count. The Gaussian Function is used for computing the conditional distribution of continuous data. The Gaussian Function's equation looks like this:

VI. THE FINDINGS AND DISCUSSION

By calculating the precision level using the Confusion-Matrix, it is possible to test the outcomes using the Naive Bayes approach. Based on both actual and prognostic data, a total of 3,122 data were employed for testing. Normal and anomaly probabilities will be employed. The traffic flow is considered to be typical if it is idle or browsing-related. When the flow of traffic is threatened, there is an anomaly. The precision of the Naive Bayes approach may be ascertained from these results.

The Naive Bayes Method's Confusion Matrix Results:

This section will go over the confusion matrix's findings after applying Naive Bayes to all test data. The confusion-matrix is used in the Accuracy computation. Using a confusion matrix is a technique that is used to determine how accurate the idea of data mining is. The following table shows the confusion matrix's findings:

TABLE I. THE RESULTS OF THE CONFUSION MATRIX

Confusion Matrix		Predicted	
		Positive	Negative
Actual	Positive	1365	1758
	Negative	8	1777

Several values can be named from the confusionmatrix table: a. True Positive (TP). Both the prediction and the data are typical. The TP data is 1365 b, as shown in Table 1. False Positive (FP). The real data is an anomaly even though the projected data is typical. Table 1 indicates that the FP data is 8.

False negative (c) (FN). The real data is normal, but the anomaly prediction data is different. Table 1 indicates that the FN data is 1758.

Truly Negative (TN). Anomalies are present in both the real data and the projected data. Table 1 indicates that the TN data is 1777.

The Naive Bayes method's accuracy, recall, precision, and error rate findings. We can obtain the numbers for recall, precision, error rate, and accuracy from the confusion- matrix table. The table that follows shows the outcomes of these values.

TABLE II. SHOWS THE FIGURES FOR ACCURACY, RECALL, PRECISION AND ERROR RATE

Recall (%)	Precision (%)	Error rate (%)	Accuracy (%)
97.50	52.27	37.98	66.02

- a) The Recall is the proportion of correctly predicted outcomes to all correct data. The recall's outcome is 99.50%. These findings show that, when compared to the complete dataset, which is 97.50%, the model was successful in retrieving information to categorize anticipated data as normal.
- b) Precision is the proportion of correctly predicted positive outcomes to all positive outcomes. The precision's outcome is 52.27%. These findings show that there is a 50.27% correlation between the requested data and the expected results provided by the model for the classification of the data that are in fact normal over the full dataset. The ratio of incorrect predictions—both positive and negative—to the total set of data is known as the error rate. 37.98% was the outcome of the error rate. These findings show that the model's accuracy in correctly classifying data that was expected to be normal or anomalous throughout the entire dataset was 37.98%. The ratio of accurate predictions (both positive and negative) to the total data is known as accuracy. The accuracy figure was 64.02%. These findings show that the model's overall classification accuracy for correctly classifying data that was anticipated to be normal or anomalous was 66.02%.

VII. CONCLUSION AND UPCOMING PROJECTS

In addition to being helpful in the automation process, interconnection networks on IoT are susceptible to network- based attacks and interruptions like DoS. This research tries to categorize DoS detection systems in a

network, but especially for IoT Raspberry Pi devices. Training and test datasets are required while building classifiers. A training dataset from NSLKDD in KDD99 format is used in this investigation. The Raspberry Pi 3 device's IDS log process is where the testing data is obtained.

Although NSLKDD is one of the frequently used datasets, it is not especially utilized for Internet of Things (IoT) devices. For some recommendations that can be made it is advised that researchers conduct a system analysis by applying a technique besides Naive Bayes in their next work if they intend to construct a system, in addition to assessing the consistency of the outcomes. It is required to conduct training datasets using datasets other than NSLKDD because, in order to achieve a high accuracy when categorizing testing data, the training-set must be able to accurately reflect the state of the testing data. Only training datasets from system simulations are used in this study. The recommendations for additional study include devising methods for obtaining training datasets from significant businesses that specialize in datasets.

REFERENCES

- [1] A W Atamli and A Martin 2020, "Threat-based security analysis for the internet of things" Proceedings - 2014 International Workshop on Secure Internet of Things, SIoT 2014
- [2] M S Asang & I Sembiring 2017, "Keamanan Data Pada Perangkat Internet Of Things Menggunakan Metode Public-Key Cryptography Teknol". Informasi-Aiti 2017
- [3] R Rizal 2018, "Network Forensics Untuk Mendeteksi Serangan Flooding Pada Perangkat Internet of Things (IoT)" J. Teknol. dan Sist. Komput. 67
- [4] R Kanagalakshmi and V N Raj 2014, "Network Intrusion Detection Using Hidden Naive Bayes Multiclass Classifier Model" 76–84
- [5] N B Amor, S Benferhat and Z Elouedi 2004, "Naive Bayes vs decision trees in intrusion detection systems" Proc. ACM Symp. Appl. Comput. 1 420–4
- [6] A Prasetyo, L Affandi and D Arpandi 2018, "Implementasi metode naive bayes untuk intrusion detection system (ids)" J. Inform. Polinema 4 280–4
- [7] S Mukherjee and N Sharma 2012, "Intrusion Detection using Naive Bayes Classifier with Feature Reduction Procedia Technol". 4 119–28
- [8] L Dhanabal and S P Shantharajah 2015, "A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms" Int. J. Adv. Res. Comput. Commun. Eng. 4 446–52
- [9] Y Wahba, E ElSalamouny and E Taweel G 2015, "Improving the Performance of Multi-class Intrusion Detection Systems using Feature Reduction"
- [10] S El-Sappagh, A S Mohammed and T A AlSheshtawy 2019, "Classification Procedures for Intrusion Detection Based on Kdd Cup 99 Data Set" Int. J. Netw. Secur. Its Appl. 11 21–
- [11] M K Siddiqui and S Naahid 2013, "Analysis of KDD CUP 99 Dataset using Clustering based Data Mining" Int. J. Database Theory Appl. 6 23–34
- [12] A Abubakar, B Pranggono, "Machine Learning Based Intrusion Detection System for Software Defined Networks", Emerging Security Technologies (EST) IEEE 978-1-5386, April-2017
- [13] Shenfield, D. Day, A. Ayesha, "Intelligent intrusion detection systems using artificial neural networks", ICT Express 2021.04.003
- [14] S Kusriani 2017, "Algoritma K-Means untuk Diskretisasi Numerik Kontinyu Pada Klasifikasi Intrusion Detection System Menggunakan Naive Bayes" Konf. Nas. Sist. Inform. 61–6
- [15] V Gustavsson 2019, "Machine Learning for a Networkbased Intrusion Detection System" examensarbete inom elektronik och datorteknik, grundnivå, 15 hp stockholm, sverige 2019
- [16] H Chae, Jo B, S Choi and T Park 2018, "Feature Selection for Intrusion Detection using NSL-KDD" Recent Adv. Comput. Sci. 20132 184–7
- [17] El-Sappagh, A S Mohammed and T A AlSheshtawy 2019 Classification Procedures for Intrusion Detection Based on Kdd Cup 99 Data Set Int. J. Netw. Secur. Its Appl. 2019.