University of Baghdad
College of Engineering
JE
JOURNAL OF ENGINEERING

Journal of Engineering

journal homepage: www.joe.uobaghdad.edu.iq

Volume 29 Number 2 February 2023

An Empirical Investigation on Snort NIDS versus Supervised Machine Learning Classifiers

Sarah Abdulrezzak *
M.Sc. student
Collage of Engineering, University of Baghdad
Baghdad, Iraq
sarah.abdulbaqi2005m@coeng.uobaghdad.edu.iq

Firas A. Sabir
Assist. Prof.
Collage of Engineering, University of Baghdad
Baghdad, Iraq
firas.a.saber@coeng.uobaghdad.edu.iq

ABSTRACT

With the vast usage of network services, Security became an important issue for all network types. Various techniques emerged to grant network security; among them is Network Intrusion Detection System (NIDS). Many extant NIDSs actively work against various intrusions, but there are still a number of performance issues including high false alarm rates, and numerous undetected attacks. To keep up with these attacks, some of the academic researchers turned towards machine learning (ML) techniques to create software that automatically predict intrusive and abnormal traffic, another approach is to utilize ML algorithms in enhancing Traditional NIDSs which is a more feasible solution since they are widely spread. To upgrade the detection rates of current NIDSs, thorough analyses are essential to identify where ML predictors outperform them. The first step is to provide assessment of most used NIDS worldwide, Snort, and comparing its performance with ML classifiers. This paper provides an empirical study to evaluate performance of Snort and four supervised ML classifiers, KNN, Decision Tree, Bayesian net and Naïve Bays against network attacks, probing, Brute force and DoS. By measuring Snort metric, True Alarm Rate, F-measure, Precision and Accuracy and compares them with the same metrics conducted from applying ML algorithms using Weka tool. ML classifiers show an elevated performance with over 99% correctly classified instances for most algorithms, While Snort intrusion detection system shows a degraded classification of about 25% correctly classified instances, hence identifying Snort weaknesses towards certain attack types and giving leads on how to overcome those weaknesses.

Keywords*:* NIDS, Snort, KNN, Decision Tree, Naïve Bays, Weka

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2023.02.11

Article received: 5/8/2022
Article accepted: 21/9/2022
Article published: 1/2/2023

تحليل اداء نظام كشف الاختراق الشبكي Snort بالمقارنة مع مصنفات التعلم الالي

سارة عبدالرزاق عبدالباقي	فراس علي صابر
طالب ماجستير	استاذ مساعد
كلية الهندسة جامعة بغداد	كلية الهندسة جامعة بغداد

الخلاصة

مع الاستخدام المتزايد للخدمات المتعلقة بالشبكات، اصبحت الامنية نقطة مهمة لجميع انواع الشبكات ولهذا ظهرت مجموعة من المنظومات لحماية الشبكات منها انظمة كشف الاختراق. وعلى الرغم من وجود العديد من انظمة كشف الاختراق الشبكي التي لاتزال فعاله في ايجاد الاختراقات الا انها من الممكن ان تحتوي على عدة مشاكل مثل كثرة الانذارات الخاطئة وعدم القدرة على اكتشاف انواع معينة من الهجمات ،مما دفع بعض الباحثين لخلق انظمه مستندة على تقنيات التعلم الالي لكي تقوم بكشف الاختراقات تلقائيا، فيما كان الاتجاه الاخر هو تسخير تقنيات التعلم الالي لتحسين انظمة الكشف الحالية ولذلك لانتشارها الواسع بين اوساط المستخدمين. لتطويرانظمة الكشف الحالية لابد من توفر تحليل موسع لمعرفة نقاط ضعفها ،بدءا بتقييم اداء نظام الكشف الشبكي Snort الاكثر استخداما بالعالم ، ومقارنة ادائه مع مصنفات التعلم الالي .يوفر هذا البحث دراسة عملية لتقييم اداء نظام Snort مع خوارزميات التعلم الالي KNN, Decision Tree, Bayesian net و Naïve Bays في كشف الهجمات الرئيسية ضد الشبكات وهي مراقبة الشبكة وكشف كلمات السر و حظر الخدمة DoS؛ يتم التقييم عن طريق احتساب معايير الدقة ، نسبة القيود المصنفة بصورة الصحيحة ، True Alarm Rate و F−measure لنظام الكشف Snort ومقارنتها بنفس المعايير المحتسبة لمصنفات التعلم الالي باستخدام اداة Wekaولنفس بيانات التناقل الشبكي . اظهرت نتائج التجربة ان نسبة القيود المصنفة بصورة صحيحة تعدت ٩٩٪ لمعظم خوارزميات التعلم الالي بينما كانت نسبة القيود المصنفة بصورة صحيحة بنظام الكشف Snort حوالي ٢٥٪ من بيانات التناقل الشبكي. اشارت الدراسة الى نقاط ضعف نظام Snort في كشف انواع معينة من الهجمات، ما يقود الى دلائل عن كيفية معالجة هذا الضعف.

الكلمات الاساسية : اداء نظام كشف الاختراق الشبكي، Snort ، KNN، Decision Tree ، Naïve Bay، Weka

1. INTRODUCTION

The need for NIDS (Network Intrusion Detection System) emerged from the vulnerabilities in traditional network that make network security questionable and were frequently exploited by hackers like the lack of reliability in Web application. The main purpose of NIDS is simply to provide an extra layer of protection in network security. By employing intrusion detection techniques, information of known attacks can be utilized to Figure out potential intrusions on servers and hosts **[Sicato, et al., 2020].** Besides, collected information from

the whole process can be used for reconnaissance purposes to analyze those attacks on deeper level and eventually update the corresponding countermeasures **[Mazurczyk and Caviglione, 2021]**.there are various NIDSs nowadays, but the most prevalent systems are Snort, Suricata and Zeek (formerly Bro). Despite their myriad popularity there are countable deficiencies in their performance, high false alarm rate and several attacks are not even detected. To improve cybersecurity in general many researchers utilized Machine learning techniques, an application of Artificial Intelligence that utilizes mathematical data models to assist a computer learns without explicit directions, to build softwares that automatically learn from data without human interaction, this is where ML classifiers surpass traditional NIDS **[Shetty, et al., 2020].** Early efforts to enhance intrusion detection system through ML algorithm were shown in 2007 by **[Tesink, 2007],** he proposed a multistage model to preprocess and classify datasets using eager learning (Repeated Incremental Pruning to Produce Error Reduction RIPPER algorithm) and lazy classifier (KNN IB1) and hybrid learners to achive elevated accuracy, recall and precision. In **[Saboor, 2013]** a thorough experiment on Snort IDS under different test benches was held to provide better performance against DDoS attacks, they proved that a better hardware will improve packet handling capacity of snort in single core machines but has no impact on the detection accuracy of DDoS attack. An evaluation of Websites Vulnerabilities Using Snort NIDS against three specific attacks is carried out in **[Dabour, et al.,2013];** they improved websites protection by providing rules for Snort to detect SQL Injection Attacks, XSS (cross side scripting) Attack, and Command Execution, those rules were tested on DVWA a vulnerable web application used for research purposes. In both **[Alqahtani, et al., 2020]** and **[Mohammed and Hussein, 2022]** several machine learning algorithms (Bayesian Network, Naive Bayes, Random Forest, Decision Tree, Random Tree, Decision Table, KNN and Artificial Neural Network) are tested with KDD99 dataset and four metrics are calculated, Accuracy, TP Rate, Precision and F-measure, the study shows that the all seven ML algorithms performed well with four metrics generally between 90% and 99% but KDD99 was released two decades ago and in term of cybersecurity it is considered outdated dataset. Finally, a detailed comparison between the most popular rule-based IDSs, Snort and Suricata was performed in **[Shah and Issac, 2018; and Isa, et al., 2019]** regarding hardware requirements, specifications, CPU usage, operating systems, methodology, and detection rates against various attack types. Yet none of these researches conducted an evaluation of Snort performance regarding detection accuracy and compare it with machine learning classifier to identify where these classifiers surpassing Snort; that is where the need for this study originated. This paper presents an assessment of Snort NIDS with the same metrics used to test machine learning algorithms and shed light on Snort weaknesses towards specific attacks. The rest of the paper is organized as follows. The remnants of Section 1 are definition of Snort NIDS, the types of ML and the metrics used to evaluate the performance. In Section 2 research method and emulation of the experimental network are described. Section 3 explains the conducted results and analyzes classifier's performance; finally, Section 4 concludes experiment's output and the related future work.

1.1 Snort

Snort is an open source, rule-based network intrusion detection and prevention system with Command Line Interface (CLI). It was Created by Sourcefire then owned by Cisco in 2013 and who is responsible for its maintenance till present time. Snort is installed in a considerable number of machines and it supports both Windows and Linux **[Dutta, et al., 2022].** Snort can run in three different modes; packet sniffer, packet logger and network intrusion detection and prevention system. In packet sniffer mode, the incoming traffic will be monitored and displayed on the console. In packet logger mode, packets will be logged to a log file that is predefined by the user. When used in intrusion detection mode the packets will be inspected by a detection engine against a reference model that defines the abnormal traffic **[Kurundkar, et al., 2012].** Snort architecture is shown in **Fig.1**. It consists of the following logical components: A Sniffer to capture traffic packets, Packet Decoders or Preprocessor which aggregate and extract feature from packets, Detection Engine that compare the extracted features against a Rule sets, most of these rules are made by Sourcefire but users can make their own custom rules as well, and finally an output model that initiates actions according to the running mode **[Hussain and Sharma, 2019]**.

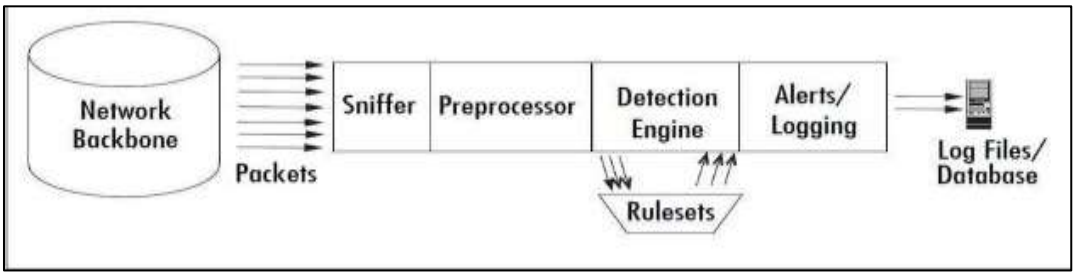

Figure 1: Snort Architecture **[Shah and Issac, 2018]**.

1.2 Machine Learning and its Algorithms

Machine learning algorithms use training data sets to develop a cognitive model that can give prediction for a new data set without any prior gaudiness. ML techniques are utilized in improving cyber security and discovering automated attacks **[Solanki, et al., 2020].** There are three basic machine learning categories based on their methodology: supervised, unsupervised, and semi-supervised machine learning. In supervised machine learning, also called task driven mechanism, the portion of data that is targeted for training are labeled and the labels are used in classification and regression computations. Unsupervised machine learning focuses on discovering relationships between instances and finding patterns among data, therefore it is known as data driven mechanism **[Shaukat, et al., 2020].** Semi-supervised ML is a combination of supervised and unsupervised methods it is used when only a small amount of data is labeled. In general, labeling the data by human experts will surely elevate model's accuracy. Therefore, classification based-supervised Machine learning algorithms are the most popular techniques used in cyber security in general and especially in intrusion detection systems **[Shaukat, et al., 2020].**

1.3 Performance evaluation Metrics and confusion matrix

Measuring the performance of an intrusion detection system is a challenging task due to the various factors that affect it. Regarding the classification process of a NIDS, the use of confusion matrix and its metrics is practical and efficient. Confusion matrix records the prediction results of a classification model and displays its performance. A confusion matrix is an outcome table with four categories:

- False positive (FP) is when normal traffic packets are misclassified as attacks.
- True positive (TP) is malicious traffic that detected correctly by the classifier.
- False negative (FN) is actual attack packets misclassified as normal traffic.
- True negative (TN) is when normal traffic classified and predicted correctly **[Barot, et al., 2014].**

There are other measures that can be conducted from the above metrics including:

- Accuracy is fraction of correctly classified instances to the overall dataset. Accuracy is the most common metric used for classification assessment and it is given by TP+TN/ (TP+TN+FP+FN).
- Precision is the fraction of true positive to all instances classified as malign. It is calculated by TP/ (TP + FP).
- True Positive Rate (Recall) is the proportion of true positives among all actual malignant instances. It is evaluated by TP Rate= TP/ (TP + FN). The difference between Precision and Recall is shown in **Fig.2.**

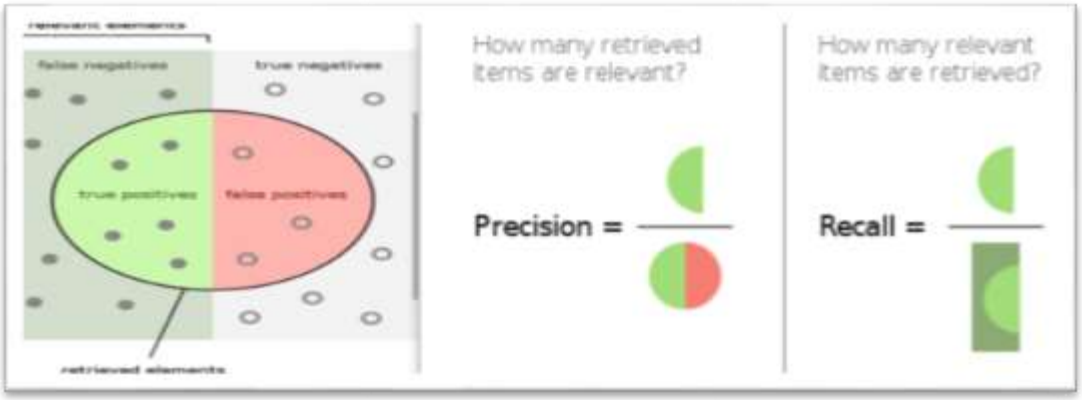

Figure 2: Precision and Recall metrics **[Maleki, et al., 2020].**

- F-Measure (F-Beta Measure): is the weighted harmonic mean of the recall and precision metrics where Beta is a factor that controls the weight of recall to precision. The formula for this metric is
  F-Beta Measure = $(1+ß^2)$ * Precision * TP Rate / ($ß^2$* Precision) + TP Rate
  The default value of Beta is1 giving an equal weight for recall and precision. In this experiment the default value is employed so the formula used for computing is 2(TP Rate* Precision) / (TP Rate+ Precision) **[Kim, et al., 2018]** and **[Hameed, et al., 2021].**

2. METHOD

To perform an empirical assessment for any classifier model; set of predefined steps are carried out as shown in **Fig.3**. A diverse dataset must be prepared and labeled with the right class type. To generate a viable dataset of this study, traffic of three attack types along with normal traffic packets are streamed in an emulated network and captured via Wireshark, a packet capturing tool to collect and analyze packets of traffic stream **[www.wireshark.org].** Preprocessing data may include sampling data, labeling, aggregation, feature selection and preparing file type for the classification model. Lastly, various evaluation metrics are measured upon classification completion. Multiple parameters of this experiment are recorded in **Table 1.**

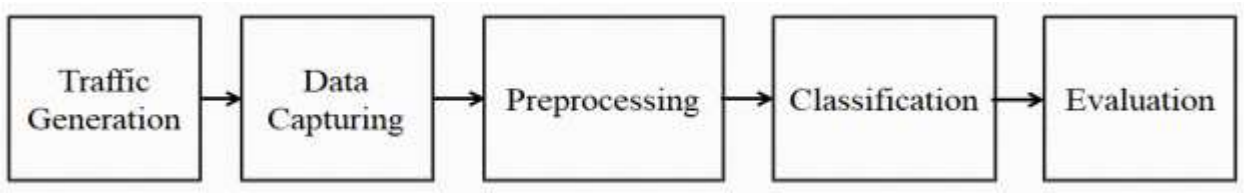

Figure 3: Performance Evaluation Method

Table 1: experiments' parameters

parameter	value
Number of packet in generated stream	2,842,378 packets
Number of sampled files	15 samples
Size of each sample	19.5 Megabytes
Number of protocol types in the stream	9 types

To build the experimental environment, an emulated network is constructed with three virtual Linux machines (Ubuntu desktop 14.4) by using Oracle VM Virtual Box V6.1 as illustrated in **Fig.4.** First machine is the attacker where three attack types are launched against the victim server; those attacks are Probing, DoS and HTTP Brute attack. The second machine is the victim server, Apache server and My SQL server and Wireshark are installed on this server. Wireshark is set to capture the traffic stream during the attacks. On the third machine the IDS Snort V2.9.18 is installed along with both community and latest registered Snort rules from Snort official site **[www.snort.org],** those rules are imposed on the PCAP (packet capture) file captured by Wireshark. Snort processes the traffic stream in intrusion detection mode with all rules enabled, any packet that matches with ruleset in the Snort detect engine will fires an alert. Output alerts are logged into a specified log file. Snort's confusion matrix is calculated manually from collected log files and resulting output.

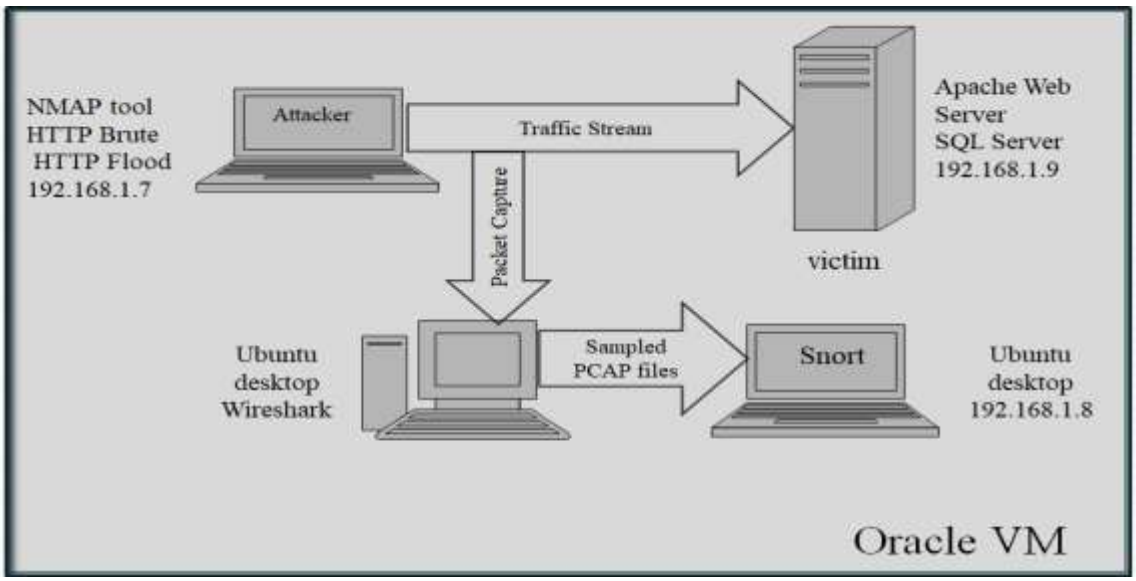

Figure 4: proposed network environment for Snort Evaluation

For the evaluation of machine learning algorithms, the same captured traffic is employed; Sampled PCAP Files are labeled manually and transformed to .arff extension (attribute relation file format) Then the implementation of machine learning algorithms is done via an open source java based tool called Weka (Waikato Environment for Knowledge Analysis); Weka is a data mining tool with both GUI and CLI interfaces and various machine learning algorithms and multiple functions including; preprocessing, clustering, regression, classification, and visualization **[www.cs.waikato.ac.nz/ml/Weka].** The confusion matrix and its related metrics are automatically calculated by this tool. Weka version 3.8.5 is installed on a windows machine where arff file is classified by four ML algorithms (KNN, Bayes Net, Decision Tree and Naïve Bayes) as shown in **Fig.5**.

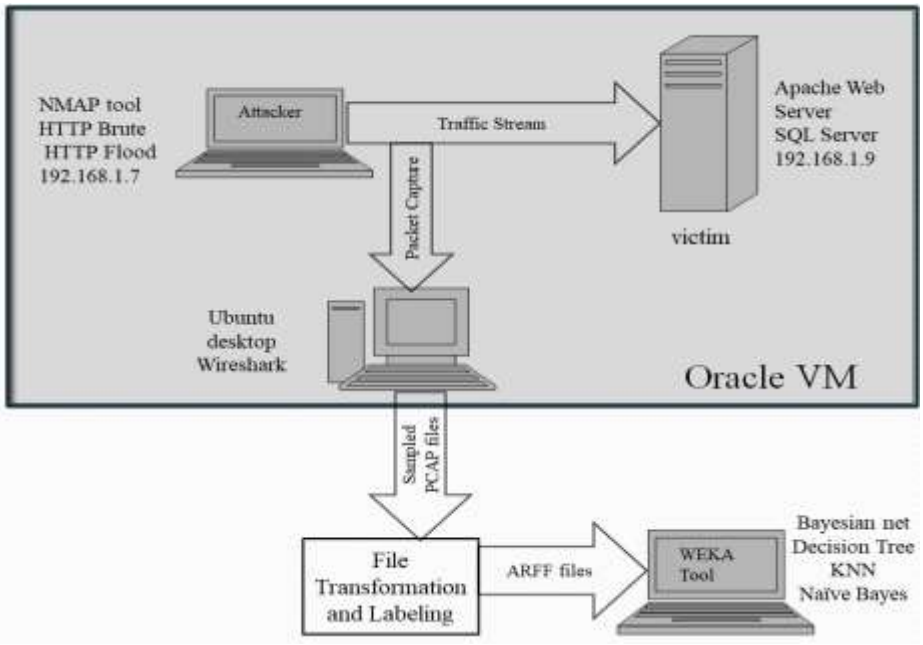

Figure 5: proposed network environment for Machine Learning Evaluation

Finally, the evaluation metrics of Machine learning algorithms are compared with the results obtained from Snort; all the findings are registered and discussed in section three. **Fig.6** sums up all the stages carried throughout the whole empirical procedure.

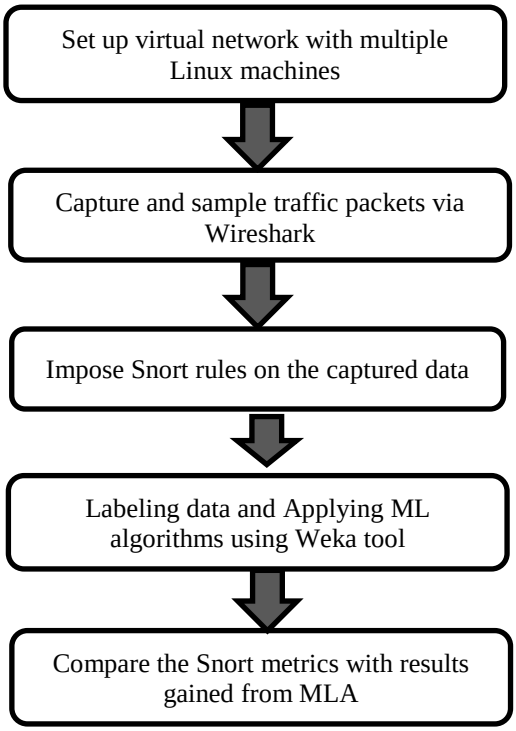

Figure 6: Stages of the Empirical Procedure

a. Type of Attacks used

Three different types of attacks are launched from the attacker machine toward the victim server. An individual packet capture file was produced for each attack type. Below is a brief description of each type.

i. Denial of Service (DoS)

DoS attack aims to shut down a computer or a process in a computer by exploiting infrastructures and network vulnerabilities to disrupt the authorized access **[Khudhur, and Croock, 2021].** H-Ping SYN-FLOOD is launched from the attacker machine against the victim server in the emulated network, flooding the attacked hosts with fake connections and spoofed information and thus preventing legitimate users from accessing targeted services and resources **[Mualfah and Riadi, 2017].**

ii. Probing

Also called scan attacks, because the attacker scans the entire network looking for available information about the open ports and services **[Laqtib, et al., 2020].** The gathered information is used for launching further attacks against the targeted entities **[Yousif and Fadahl, 2021].** NMAP is a common network mapping tool, it was installed in the attacker machine and used for initiating probing attacks against the victim server. There are numerous kinds of probing attacks including TCP scan, port scan or host scan, UDP scan SYN Stealth, and network scanning (searching for service hosts on particular ports), ACK scan, and traceroute **[Khamphakdee, et al., 2014],** all of them are initiated using NMAP tool commands.

iii. Brute force attacks

Brute force attacks involve a variety of hacking techniques that employ password guessing to gain access to a system. Numerous amounts of passwords or passphrases are submitted by the attacker in the anticipation of accurately guessing it **[Bharati, et al., 2017].** In this experiment, HTTP-Brute attack initiates brute force password auditing against http authentication of the apache server of the victim host and guesses its credentials successfully.

b. Supervised machine learning algorithms

As it was established by previous studies, Classification algorithms are the most feasible machine learning techniques for detecting intrusion. Below are descriptions of the ML classifiers applied in this empirical study.

i. Decision Tree (DT)

Decision Tree is a supervised machine learning algorithm build on a recursive tree structure. The main components of the DT are root, intermediate and leaf node, and a divergent path that binds the root node with leaf nodes **[Saeh, et al., 2016].** Every path in the tree displays all the possible values of the root node. The classified attribute is translated by the leaf node. If-then patterns are used to represent the resulting tree **[Abbas and Kareem, 2018].** Entropy and information gain are measured and used to identify the optimal intermediate node during tree generation **[Grąbczewski, 2014].** C4.5 is a prominent algorithm of decision tree; it handles the problem of over fitting via tree pruning technique **[Wattanapongsakorn, et al., 2012].** J48 is an implementation of C4.5; it was used via Weka tool to classify the labeled data file of this experiment.

ii. K-Nearest Neighbour

This classifier assumes that data points with shared similarities are closer to each other in the space than those with no similarities **[Aburomman, and Reaz, 2016**]. Lazy IBK, a KNN algorithm that processed the arff data file, utilizes a distance metric to find k "close" instances in the training data for each test instance, then makes a prediction based on those instances. Generally, KNN classifiers require less time to train than other classifiers but throughout the classification process, its computation time is an overhead.

iii. Bayesian Network and Naive Bayes

Bayesian Network classification of probability distribution is built upon conditional independencies. Bayes net or Bayesian Networks can be utilized for reasoning, diagnosis, and detection. Naive Bayes, a type of Bayesian network, predicts attack type in the dataset by calculating its probability. Naive Bayes was considered to be fast relatively to other three classifiers. Conditional probability and class probability are both used by Naive Bayes algorithm. Class probability is the ratio of the occurrence of each class instance to the total number of instances. Conditional probability is calculated by dividing the frequency of each attribute in a specific class to the number of samples for the same class. Both Bayes Net and Naive Bayes were implemented via Weka tool.

3. RESULTS AND DISCUSSION

The confusion matrix and its metrics, F-Measure, Accuracy, Precision, True Positive Rate and Percentage of correctly classified instances, were calculated automatically by Weka tool for the Machine Learning algorithms (KNN, Bayes Net, Decision Tree and Naïve Bayes), each run separately, with 60% of the sampled data used for training the classifiers and the remaining 40% used for testing them. These results were recorded in **Table 2** along with metrics obtained manually from Snort output logs. All the metrics are conducted with the same captured data file. Decision tree, KNN and Bayes Net had the best performance in Precision, Accuracy, True Positive Rate, F-Measure and with over 99% correctly classified instances, naïve Bayes came second with over than 75% accuracy and finally Snort accuracy with 0.25

TP Rate, F-Measure equals 0.385, 0.842 Precision, 0.477 Accuracy and almost 25% correctly classified instances which is considered quite degraded metrics comparing with ML algorithms score**. Figs.7 to 11** depict bar charts of F-Measure, Accuracy, Precision, True Positive Rate (Recall) and Percentage of correctly classified instances for the selected MLA and Snort, in the same order. confusion matrix of Snort performance are calculated manually for the samples of traffic and represented by **Table 3**, the confusion matrix shows that about 75% of the malicious traffic where falsely predicted as normal traffic, which affects Snort performance extremely. In **Table 4** True Positive Rate, F-Measure, Precision, Accuracy and the percentage of correctly classified instances of Snort NIDS are all calculated for each attack type to provide a better understanding of the reason behind degraded prediction. As shown in **Table 4** only 29.6% of DoS attacks packets are detected while brute force attacks is not detected at all**.**

Table 2: TP Rate, F-Measure, Precision, Accuracy and the percentage of correctly classified instances for the selected Machine Learning Algorithms and Snort IDS

ML Algorithm	TP Rate	F-Measure	Precision	Accuracy	correctly classified instances
Naive Bayes	0.751	0.687	0.803	0.757	75.1473%
Decision Tree	0.996	0.995	0.996	0.997	99.5575%
KNN	0.996	0.995	0.996	0.998	**99.6036%**
Bayes Net	0.996	0.995	0.996	0.997	99.5659%
Snort	0.250	0.385	0.842	0.477	24.9393%

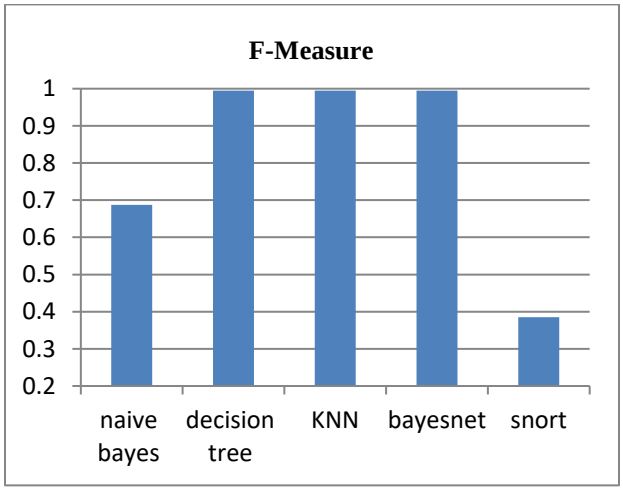

Figure 7. F-Measure for selected MLA and Snort

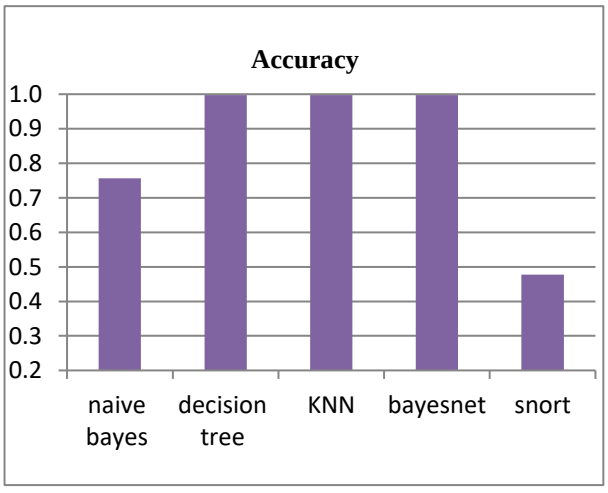

Figure 8. Accuracy for the selected MLA and Snort

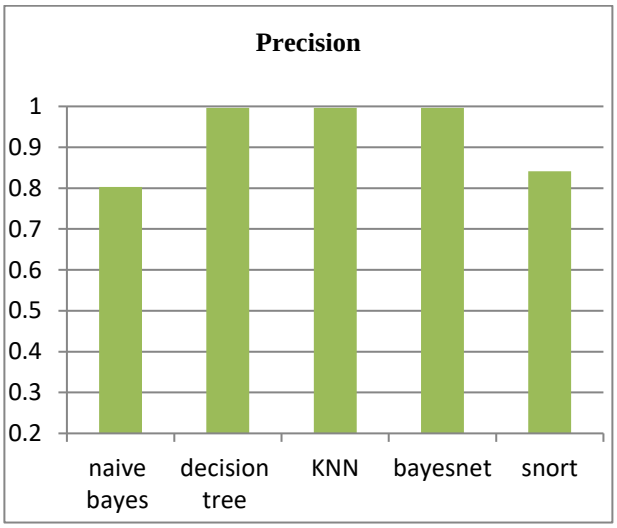

Figure 9. Precision for the selected MLA and Snort

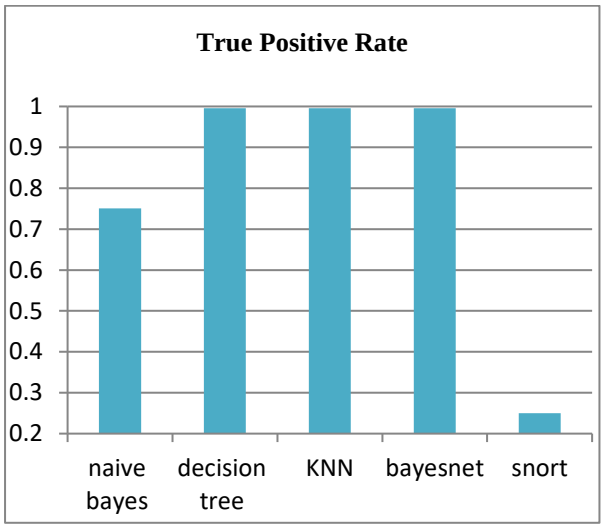

Figure 10. True Positive Rate for the selected MLA and Snort

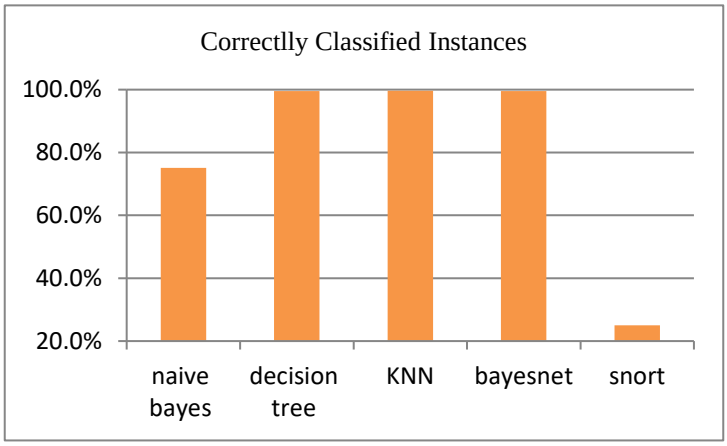

Figure 11. Percentage of the correclt classified instances for the selected MLA and Snort

Table 3. Confusion Matrix of Snort Detection

		predicted traffic			
		DoS	HTTP Brute	NMAP	Normal
Actual traffic	DoS	706775	0	0	1682590
	HTTP Brute	0	0	0	450123
	NMAP	0	0	2186	561
	Normal	0	0	3	151

Table 4. TP Rate, F-Measure, Precision, Accuracy and the percentage of correctly classified instances of snort for each attack type

Attack type	TP Rate	F-Measure	Precision	Accuracy	Percentage of correctly classified instances
DoS	0.296	0.4566	1.00000	0.40804	29.57%
HTTP Brute	0.000	0.0000	0.00000	0.84164	0%
NMAP	0.796	0.8857	0.99890	0.99980	79.59%
Normal	0.981	0.0001	0.00007	0.24948	98.08%

4. CONCLUSIONS

The collected metrics from the above section show that despite the provided up to date Snort rules, there are still some drawbacks in the detection accuracy of Snort NIDS unlike Decision Tree, KNN and Bayes Net classifiers, where detection accuracy is generally above 99 percent. In Snort NIDS, normal traffic and probing attack packets were predicted by over 88% on average, but, over two thirds of DoS attack packets were misclassified as normal traffic and failed to detect the entire traffic of http brute force attack. This leads to the fact that, Snort performance require additional enhancement and Snort rules can use further updating to include those two attacks in specific. A futuristic extension for this work might include other attack types and tools plus a vigorous analysis on attributes of DoS and Brute force captured traffic data file including feature extraction and selection along with a detailed examination of machine learning classification mechanism will be the next step to upgrade the performance of Snort NIDS achieve higher detection accuracy and reduce the gap between ML classifiers and Snort.

REFERENCES

Sicato, Jose Costa Sapalo, Sushil Kumar Singh, Shailendra Rathore, and Jong Hyuk Park, 2020. "A comprehensive analyses of intrusion detection system for IoT environment." Journal of Information Processing Systems, 16 (4), pp. 975-990.

Mazurczyk, Wojciech, and Luca Caviglione, 2021. "Cyber reconnaissance techniques." Communications of the ACM, pp: 86-95.

Shetty, Nisha P., Jayashree Shetty, Rohil Narula, and Kushagra Tandona, 2020. "Comparison study of machine learning classifiers to detect anomalies." International Journal of Electrical and Computer Engineering (IJECE), 10 (5), pp. 5445.

Tesink, Sebastiaan, 2007, "Improving intrusion detection systems through machine learning." Group (07). www.bughunt.org/ thesis_lai.pdf.

Saboor A, Akhlaq M, Aslam B. 2013, "Experimental evaluation of Snort against DDoS attacks under different hardware conFigureurations", National Conference on Information Assurance (NCIA), IEEE, pp. 31-37.

Dabbour, Mohammad, Izzat M. Alsmadi, and Emad Alsukhni. 2013, "Efficient assessment and evaluation for websites vulnerabilities using SNORT." International Journal of Security and Its Applications.7(1).

Alqahtani, Hamed, Iqbal H. Sarker, Asra Kalim, Minhaz Hossain, Syed Md, Sheikh Ikhlaq, and Sohrab Hossain. 2020, "Cyber intrusion detection using machine learning classification techniques." In International Conference on Computing Science, Communication and Security, pp. 121-131.

Hussein M. and Mohammed S., 2022. "Performance Analysis of different Machine Learning Models for Intrusion Detection Systems." Journal of Engineering 28(5), pp. 61-91.

Shah, Syed Ali Raza, and Biju Issac. 2018, "Performance comparison of intrusion detection systems and application of machine learning to Snort system." Future Generation Computer Systems 80, pp. 157-170.

Isa, F.M., Saad, S., Fadzil, A.F.A. and Saidi, R.M., 2019. "Comprehensive performance assessment on open source intrusion detection system. " In Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017) Springer, Singapore, pp. 45-51.

Dutta, Nitul, Nilesh Jadav, Sudeep Tanwar, Hiren Kumar Deva Sarma, and Emil Pricop. 2022, "Intrusion Detection Systems Fundamentals." In Cyber Security: Issues and Current Trends, pp. 101-127.

Kurundkar, G. D., N. A. Naik, and S. D. Khamitkar. 2012. "Network intrusion detection using Snort." International Journal of Engineering Research and Applications 2, pp. 1288-1296.

Hussain, Abid, and P. Sharma, 2019,"Efficient Working of Signature Based Intrusion Detection Technique in Computer Networks", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), pp.60-64.

Solanki, S., C. Gupta, and K. Rai. 2020 "A Survey on Machine Learning based Intrusion Detection System on NSL-KDD Dataset." Int. J. Comput. Appl 176 ,pp: 36-39.

Shaukat K, Luo S, Chen S, Liu D.,2020, "Cyber threat detection using machine learning techniques: A performance evaluation perspective", International Conference on Cyber Warfare and Security (ICCWS), IEEE, Oct 20, pp:1-6.

Shaukat, Kamran, Suhuai Luo, Vijay Varadharajan, Ibrahim A. Hameed, and Min Xu. 2020, "A survey on machine learning techniques for cyber security in the last decade." IEEE Access 8, pp. 222310-222354,

Barot, Virendra, Sameer Singh Chauhan, and Bhavesh Patel. 2014. "Feature selection for modeling intrusion detection." International Journal of Computer Network and Information Security, pp. 56-62.

Maleki, F., Ovens, K., Najafian, K., Forghani, B., Reinhold, C. and Forghani, R., 2020. Overview of machine learning part 1: fundamentals and classic approaches. Neuroimaging Clinics, 30(4), pp.17-32.

Kim, Kwangjo, Muhamad Erza Aminanto, and Harry Chandra Tanuwidjaja. 2018 "Network intrusion detection using deep learning: a feature learning approach.".

Hameed, Ibtihaal M., Sadiq H. Abdulhussain, and Basheera M. Mahmmod., 2021. "Content-based image retrieval: A review of recent trends." Cogent Engineering, 8(1), pp. 1927469.
https://www.wireshark.org.

Khudhur, Dhuha Dheyaa, and Muayad Sadik Croock, 2021. "Developed security and privacy algorithms for cyber physical system." International Journal of Electrical & Computer Engineering. pp: 2088-8708.

Mualfah, Desti, and Imam Riadi, 2017."Network forensics for detecting flooding attack on web server." International Journal of Computer Science and Information Security 15.2, pp. 326.

Laqtib, Safaa, Khalid El Yassini, and Moulay Lahcen Hasnaoui. 2020. "A technical review and comparative analysis of machine learning techniques for intrusion detection systems in MANET." International Journal of Electrical and Computer Engineering. pp. 2701.

Yousif, Samar Taha, and Zaid Abass Fadahl. 2021. "Proposed Security Framework for Mobile Data Management System." Journal of Engineering, 27 (7), pp: 13-23.

Khamphakdee, Nattawat, Nunnapus Benjamas, and Saiyan Saiyod. 2014. "Improving intrusion detection system based on snort rules for network probe attack detection.". 2nd International Conference on Information and Communication Technology (ICoICT), IEEE, pp. 69-74.

Bharati, Manisha, and Sharvaree Tamane. 2017. "Defending against bruteforce attack using open source SNORT". International Conference on Inventive Computing and Informatics (ICICI), IEEE.

Saeh I, Mustafa W, Al-Geelani N., 2016. "New Classifier Design for Static Security Evaluation Using Artificial In-telligence Techniques". International journal of electrical and computer engineering.

Abbas, A. R., & Kareem, A. R., 2018. "Intelligent age estimation from facial images using machine learning techniques". Iraqi Journal of Scienc. pp. 724-732.

Grąbczewski, Krzysztof. 2014. "Meta-learning in decision tree induction." Cham: Springer International Publishing, 1.

Wattanapongsakorn N, Srakaew S, Wonghirunsombat E, Sribavonmongkol C, Junhom T, Jongsubsook P, Charnsripinyo C., 2012. " A practical network-based intrusion detection and prevention system", IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Jun 25, pp. 209-214.

Aburomman, Abdulla Amin, and Mamun Bin Ibne Reaz., 2016. "Review of IDS development methods in machine learning." International Journal of Electrical and Computer Engineering, pp. 2432-2436.