Development of an Intrusion Detection System in Web Applications Using C-Means and Decision Tree Algorithm

Rafiu Mope ISİAKA[1], Damilola David POPOOLA[1, *]
[1]Department of Computer Science
Faculty of Information and Communication Technology
Kwara State University
Malete, Nigeria

Abstract

Intrusion detection is extremely important for online applications and for determining whether there has been a hostile entrance into the website. The aim of this research is to provide a machine learning technique for detecting intrusion in a web application.

Machine learning models such as C-means, Decision Tree and Support Vector Machine were utilized to create an intrusion detection system. The study used the CIC-IDS 2018 intrusion dataset (Friday-Working Hours-Afternoon-Ddos.pcap ISCX). The data was initially sent to Decision tree and SVM which had accuracy of 99.97% and 99.77%, respectively. The raw data was next transferred into the c-means clustering approach, which had an accuracy of 99.99%. The goal of the clustering technique used is to improve the system's accuracy, and the results were assessed using performance metrics like accuracy, sensitivity, precision, specificity, F1-score as well as accuracy comparison of the results obtained with the state of the art.

Keywords: Intrusion Detection System, Web application, Machine Learning, Clustering

1. INTRODUCTION

Signature-based intrusion detection systems, anomaly-based intrusion detection systems, and hybrid intrusion detection systems are the three types of intrusion detection systems (Alyousef, 2019). Anomaly-based intrusion detection systems, as well as a variety of other analytical methodologies have recently been developed and used to track novel system threats. These tactics can achieve detection rates of 98 percent at a high alert rate and 1% at a low alert rate (Jacob et al., 2017).

The study looked at the similarities and conflicts that have emerged in the development of machine learning methods and techniques for fault detection and cybersecurity in complex network defence, as well as the need to differentiate between the two. Industrial Control Systems (ICS) intrusion detection systems are frequently trained on network packet captures and focus primarily on network layer traffic monitoring for intrusion detection. (2020, Ayodeji et al.) Machine learning techniques are used by intrusion detection systems to discover and recognize security issues.

For several applications, including fog computing, the Internet of Things (IoT), big data, smart cities, and 5G networks, intrusion detection systems use machine learning approaches. Machine learning approaches such as Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest can be used to detect infiltration. (Saranya et al, 2020b.)

Failure to recognize and distinguish between the fundamentally similar signature that distinguishes typical transients common to a complex device, incipient/slowly-developing fault, and cyber interference with physical effect on the process information is a major contributor to false alarm generation. As a result, the rate of false alarms in nuclear power plant control systems that account for observed process calculation shifts and intrusion detection systems used to identify intrusions on industrial controllers is significant. (Ayodeji et al, 2020.)

To target computer users, cybercriminals deploy sophisticated strategies and social engineering techniques. As time passes, cybercriminals get more advanced and inspired. Cybercriminals have demonstrated their abilities to hide their identities, send encrypted conversations, keep their identities separate from unlawful profits, and use secure technology. As a result, advanced intrusion detection solutions capable of identifying current ransomware are becoming more important for the security of computer systems. To plan and build such Intrusion Detection System programs, it is necessary to have a thorough awareness of the strengths and weaknesses of current Intrusion Detection System research (Khraisat et al., 2019).

This study suggests the use of a machine learning strategy to detect intrusion in a website or web application by classifying and reviewing machine learning based ways for cyber security researchers using data objects.

There are various types of attacks, but the most difficult to detect is the insider/internal attack. When it comes to network security, every user wants his machines to be safe from all hostile attacks (internal or external attacks). Internal intrusion detection and protection systems can identify internal invaders, whereas exterior intrusion detection and protection systems can detect external intruder attacks. In exchange, these strategies help us protect our systems. Borkar et al. (Borkar et al., 2018). In recent years, machine learning algorithms have been employed to identify and classify risks (Saranya et al., 2020a).

The ability of an intrusion detection system (IDS) to detect potential attacks is crucial for protecting network resources and data from the attack's destructive consequences. Among the many tactics available for insertion into intrusion detection systems to boost their accuracy, classification algorithms such as decision trees have been demonstrated to provide spectacular and efficient results in identifying assaults and require further exploration in IPv6-based attacks. Choudhury et al., 2015).

The use of k-means and decision tree techniques to improve intrusion detection in online applications is suggested in this study.

1.1 Overview of IDS

Intrusion is defined as any unauthorized operation that harms a computer system. This assures that any threat to the security, accuracy, or availability of information will be regarded an infringement. Intrusions are activities that, for example, make computing systems unresponsive to genuine consumers. An intrusion detection system (IDS) is a software or hardware device that monitors computer networks for hostile activity to maintain system stability (Khraisat et al., 2019). The numerous types of intrusion detection techniques are Signature based detection systems, Anomaly based intrusion detection systems, and Specification based detection systems.

1.2 Classification by Data Source

Intrusion detection systems based on hosts have the advantage of being able to track sensitive object behavior as well as pinpoint incursions and induce responses (for instance: complex records, plans and ports). Intrusion detection systems based on hosts have a few drawbacks, including the fact that they absorb host assets, rely on hosts, and are unable to detect attacks in networks. In most cases, network-based intrusion prevention systems are built on important hosts or switches. The majority of the network's IDSs are self-contained. (Liu, 2019) Liu, Liu, Liu, Liu, Liu

In a variety of scenarios, operating systems are utilized. Furthermore, network-based intrusion detection systems can detect specific types of protocol and network attacks. The disadvantage is that it only tracks traffic passing through a certain region of the network (Bul et al., 2015). The main distinctions between host-based intrusion detection systems and network-based intrusion detection systems are noted in Table 1.

2. RELATED WORKS

Signature-based, anomaly-based, stateful protocol analysis-based, and hybrid-based detection approaches were utilized by Mudzingwa (2014). While the anomaly-based methodology surpasses the other two in terms of detecting new risks without the need for user feedback or revisions, many users prefer the other two approaches. Existing IDPS on the market use a combination of the four major techniques. It also made evaluating and testing the IDPS methodologies used by current IDPS products much easier. Experiments with commercial and open-source software, as well as our evaluation standards, will be part of the study's future findings.

Artificial neural networks were proposed by Shenfield (2018) as a novel means of identifying malicious network traffic that might be employed in deep packet inspection-based intrusion detection systems. A variety of benign network traffic data (images, dynamic link library files, and a variety of other miscellaneous files such as logs, music files, and word processing documents) as well as malicious shell code files from online exploit and vulnerability repository exploits can be used to destitute the proposed artificial neural network architecture. The suggested artificial neural network design achieves a 98 percent average accuracy, a 0.98 average region under the receiver operator feature curve, and a less than 2% average false positive rate after repeated 10-fold cross-validation.

Sharma (2015) used machine learning techniques that have been proved to be successful in detecting intrusions. Machine learning approaches may detect intrusions with high accuracy, although the accuracy is typically influenced by other factors. Choosing the right feature set, training and testing data, and so on are just a few examples. You can improve your results by selecting the appropriate quality for these factors. Machine learning algorithms, on the other hand, may have flaws, such as distortion of network data due to poison learning.

Dey (2016) also used several machine learning methods for data mining, image processing, and predictive analytics. The major benefit of machine learning is that once an algorithm learns how to deal with data, it can do tasks independently.

Thakkar (2020) investigates datasets in the field of Intrusion Detection Systems (IDS). Based on machine learning and data mining, these datasets were utilized to assess the effectiveness of IDS. The underlying dataset should be updated, according to the findings, to better distinguish fresh assaults in the field of IDS. Because attackers use a wide range of procedures and technology in their attacks, this is the case. In addition, the process of launching various assaults duplicates the need for datasets with realistic network conditions. In the future, they'll concentrate on evaluating the output of these datasets using a variety of machine learning and data mining methodologies, as well as incorporating feature engineering and data sampling to correct the dataset's flaws.

Ayogu (2019) advocated using a decision tree to collect relevant traits, which has the advantage of providing intelligible rules. The 41 characteristics of the KDD'99 dataset were reduced to 29 using a C4.5 decision tree dimensional reduction technique. A rule-based classification system (decision tree) and a Bayesian denial of service attack (DoS) network classification system were then constructed based on the selected attributes. The results from the test dataset were used to evaluate and compare the classifiers. According to experimental data, the Decision Tree is more stable and delivers the highest percentage of effective classification than the Bayesian Network, which has been shown to be vulnerable to discretization techniques. The importance of attribute selection in the construction of a real-world intrusion detection system has been demonstrated (IDS).

Ahmad (2020) presented a taxonomy based on prominent ML and DL approaches used in the development of network-based IDS (NIDS) systems, outlining the IDS definition and then offering a taxonomy based on prominent ML and DL approaches used in the construction of network-based IDS (NIDS) systems. Examining the benefits and shortcomings of the proposed solutions provides a comprehensive overview of current NIDS-based publications. Then, in terms of recommended technique, assessment metrics, and dataset collection, recent developments and advancements in ML and DL-based NIDS are explored. We provided a few research problems and the prospective scope for investigation to construct ML and DL-based NIDS based on the shortcomings of the presented methodologies.

Recent developments in intrusion detection algorithms, as well as their weaknesses, limits, and current position in critical infrastructure applications, were explored by Ayodeji (2020). We also look at the parallels and conflicts found in the application of machine learning tools and techniques for defect detection and cybersecurity in complex system protection, as well as the need to distinguish between the two. Failure to recognize and distinguish between the fundamentally similar signatures that define normal transients common to a complex system, incipient/slowly developing fault, and cyber intrusion with physical impact on process information is a major contributor to false alarm generation, according to this study. He highlighted characteristics of nuclear plant control systems that account for observed process measurement changes, as well as a high false alarm rate for intrusion detection systems used to identify intrusions on industrial controllers, to support his position.

Bul, (2015) described an advanced software development that uses Cisco Catalyst Switches' Quality of Service (QoS) and parallel techniques to improve the analytical performance of a Network Intrusion Detection and Protection System (NIDPS) when deployed in high-speed networks, as well as designing a real network to present experiments using a Snort Network Intrusion Detection and Protection System. On the other hand, an intrusion detection and prevention system are commonly recognized as one of the most effective technologies for detecting threats and assaults. Network Intrusion Detection and Protection Systems have piqued the interest of many companies and governments, and they are accessible to anybody with Internet access. The four steps of a Network Intrusion Detection and Protection System for safeguarding a computer system network are scanning, analyzing, detecting, and repairing. The focus of our article was on the scanning and analyzing weaknesses in high-speed network connectivity by Network Intrusion Detection and Protection Systems. We advocate adopting a QoS setup and parallel technologies to improve NIDPS analysis efficiency and reduce Network Intrusion Detection and Protection System processing time. As a result of our methodology, gadgets can be designed to make thwarting attacks easier. The current and anticipated potential demands for internet security necessitate the reworking of existing systems to establish more resilient parallel systems and rule sets.

Different evolution techniques for intrusion detection systems were proposed by Almasoudy (2020). The fundamental concept is to use Differential Evolution to pick a few features from the 41 features in the NSL-KDD datasets, and then use Extreme Learning Machine to compute the accuracy of these features. Differential Evolution is used until the smallest number of high-accuracy features is found. The results indicated a higher detection rate and a decreased false alarm rate in both five and binary classification. With less training and testing time, the suggested system achieved an accuracy of 80.15 percent for five classification and 87.53 percent for binary classification. We plan to create links from live networks in the future. The U2R attack is one of the most difficult to detect in IDS because its behaviour is quite like that of standard forms, making detection difficult. The U2R attack is one of the challenges in IDS because the behaviour of this form is very similar to standard, making detection difficult. Testing the model and using a complex classifier to achieve higher detection will detect the U2R

attack, which is one of the challenges in IDS because the behaviour of this form is very similar to standard, making detection difficult.

Anomaly-based intrusion detection system was developed by Veeramreddy and Munivara (2020). It is independent of the research in a fast and sufficient pre-processing phase, which is a significant obstacle. Another problematic issue is the rise of zero-day attacks, which emphasizes the need for security systems that can accurately detect previously undisclosed threats. An attempt is made to construct a general meta-heuristic scale for both known and unknown assaults with a high detection rate and low false alarm rate using active feature optimization methodologies.

By removing the imbalanced class issue that is typically associated with network traffic datasets, Akinyemi (2019) enhanced identification accuracy. In the test case scenario with Wire shark, live network traffic packets were gathered during ordinary network activities, Sync flood assault, slow http post attack, and exploitation of known vulnerabilities on a targeted device. The Spleen tool was used to extract 52 features from the packet meta-data, including 42 features that were identical to the intrusion detection dataset from the Information Discovery in Database (KDD'99). The min-max normalization approach was used to standardize the characteristics, and the Knowledge Gain algorithm was utilized to choose the best discriminatory features from the feature space. A cascade of k-means clustering algorithm and random-forest classifier was used to create an anomalous intrusion detection model. The evaluation result showed a 10% increase in detection accuracy, a 29% increase in sensitivity, and a 0.2 percent increase in specificity when compared to the current model.

The experiment was conducted using the KDD cup 99 dataset, and Ikuomola (2015) recommended that principal Component Analysis (PCA) was utilized to decrease the characteristics in the dataset to reduce the amount of computer resources required to detect attacks. The results show that the Neural Network algorithm (Nnge) outperformed the other systems, with a minimum false positive rate of 0.9 percent and 1% before and after the function reduction, respectively. With the highest classification accuracy, the false negative alert rate was 3.2 percent before and after feature removal.

A machine learning methodology was developed by Liu and Lang (2019). The IDS taxonomy was designed with the goal of identifying machine learning-based and deep data items, with the primary goal of summarizing them. IDS literature with a learning focus. We believe that this form of classification system is suitable for cyber security researchers. Second, the survey specifies the definition of IDSs and their taxonomy. Following that, IDSs, metrics, and benchmark datasets are used to implement the most used machine learning approaches. Then, using the suggested taxonomic structure as a basis, we show how to use machine learning and deep learning methodologies to solve significant IDS challenges, with examples from the literature. Finally, current sample reports are used to analyze problems and future trends.

As a result of the effort, Mahani and Ali (2020) worked on machine learning algorithms for intrusion detection. One of the most pressing challenges in today's world is network security. As the Internet has developed dramatically and become more widely utilized over the last decade, network security vulnerabilities have become a critical problem. An intrusion detection device is used to detect illegal access and unexpected attacks over secured networks. Several studies on intrusion detection systems have been undertaken in recent years. This survey research, on the other hand, looked at 49 related papers from 2009 to 2014 that focused on single, hybrid, and ensemble classifier design architecture to better grasp the current state of machine learning.

Feature selection, various search algorithms, and attribute assessors, according to Ayo (2020), have been combined to facilitate innovation and comparability. The researchers discovered that the number of features used had no effect on the detection accuracy of function selection methods but was closely related to the output of the basic classifier. With a 1.2 percent, 98.8%, 7.17s, and 3.11s decrease in false alarm rate, high accuracy rate, and decreased training and testing time, respectively, the suggested approach outperforms other similar methods in terms of false alarm rate, high accuracy rate, and decreased training and testing time. Simulation investigations employing conventional assessment criteria also demonstrated that the proposed technique is appropriate for NIDS attack classification.

3. Study Approach

The system in this study is made up of two stages: pre-processing and classification. The dataset is loaded as an input, pre-processed using k-means clustering, the essential information in the dataset is extracted, the features are classified using two decision tree classification techniques (DT1, DT2), and the results are compared.

To split the dataset in this study, the C-Means clustering strategy is used; it is an unsupervised learning procedure for finding clusters with the nearest mean, which functions as a prototype for the cluster. A decision tree is used as

the classification mechanism. The results of this methodology are based on categorization performance. The methodologies employed in this study are broken down as follows:

(i) Use C-Means as a clustering technique on the dataset.
(ii) Use Decision tree classification performance.
(iii) Assess the accuracy, specificity, sensitivity, precision, and processing costs of the results.

Step 1- Using K-Means Clustering algorithm method; pre-process the churn customer dataset.
Step 2- Using Decision tree, as a Classification algorithm to carry out and simplify the performance of the dataset, to predict and improve intrusion level.

3.1. Experimental Dataset

The proposed system employs an intrusion detection dataset from CIC-IDS2017 to achieve the desired outcome (Friday-Working Hours-Afternoon-DDos.pcap ISCX). The dataset includes both benign and common attacks, and it closely mirrors real-world data (PCAPs). It also contains the results of a CICFlowMeter network traffic analysis, which includes flow labels based on the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack vectors (CSV files).

3.2. Clustering Algorithm

To investigate the differences in efficiency performance, C-means was employed for clustering approaches in this study. When there is unlabeled data, or data that does not contain specific categories or groupings, C-means clustering is an unsupervised learning strategy. The variable C represents the number of groups in a data collection, and the algorithm detects them. This algorithm assigns each data point to one of K groups iteratively based on the features provided.

We're interested in pointing out weaknesses and suggesting solutions to the C-means clustering algorithm, which works well with compact and hyper-spherical clusters. The focus is focused on two concerns with the C-means method that cannot be avoided:
(i) the number of clusters and centroids assigned, and
(ii) the ability to manage various types of data.
Pseudo code for k-means clustering

Function Direct-k-means ()
initialize k prototypes ($w_{1,\ldots\ldots,} w_k$) such that $w_{j=}$
i_j, $j \in \{1,\ldots\ldots,k\}$, $l \in \{1,\ldots\ldots,\}$
each cluster C_j is associated with prototype w_j
Repeat
for each input vector i_j, where $l \in \{1,\ldots..n\}$,
do
Assign i_i to the cluster C_{i*} with nearest prototype w_{j*}
(i.e, $| i_i - w_{i*} | <= | i_i - w_i |$, $j \in \{1,\ldots..k\}$)
For each cluster C_j, where $j \in \{1,\ldots..k\}$,
do
Update the prototype w_i to be the centroid of all samples currently in C_j,
So that $w_i = Zi_i \in C_i \; i_l / | C_i |$
Compute the error function:
$E = Z_{i=i} Zi_i \in c_i | i_i - w_i |^2$
Until E does not change significantly, or cluster membership no longer changes

3.3. Classification Algorithm

A Decision Tree is a graph that looks like a tree, with core nodes representing tests on attributes, branches representing test results, and leaf nodes representing class labels. The path from the root node to the leaf determines the classification rules. Because it is the most obvious attribute, the root node is chosen first to split each input data set. At each intermediate node, the tree is created by describing features and their associated values, which will be used to evaluate the input data. It can analyse data and detect important network features that indicate malicious activity. By evaluating a large amount of intrusion detection data, it can add value to many real-time security systems. It can spot patterns and trends that can help with additional study, attack signature growth, and other monitoring tasks. The fundamental benefit of decision trees over other classification methods is that they provide a comprehensive set of rules that are simple to comprehend and connect into real-time systems. (Rai et al., 2016).

4. RESULTS AND DISCUSSION

The raw data is classified with decision tree in the first phase, and the raw data is preprocessed using the clustering approach c-means in the second phase, before being passed through the various classifiers previously stated.
The results of the implementation on the jupyter notebook platform, which include 225711 characteristics and 78 attributes, are then compared. The jupyter on python google Collaboratory environment is used to implement the project, Figure 2 shows the loaded dataset.

The loaded dataset, which contains 225711 features and 78 characteristics, is loaded and imported into the jupyter IDE. To make the data useable, it is cleaned using the C-means Clustering method. This is commonly used in machine learning to partition data into train, test, and validation sets. Training and testing are two subgroups of any algorithm. The training set was utilized to fit the model and conduct assessment tests. In this project, 75% of the time was spent on training and 25% on testing.

4.1. Classifier

In this study, data is classified using decision tree; nevertheless, the data is fed into the specified classifiers, thus c-means clustered data is also passed into the classifiers, as seen in figure 3, figure 4.
The confusion matrices are accessed using the sensitivity, specificity, precision, negative predictive value, false positive rate, false accuracy rate, false negative rate, accuracy, F1 score, and Matthews correlation. Table 4.1 lists the experiment evaluation metrics.

Several tests were carried out in this inquiry, and the results are provided in table 4.1; however, table 4.2 shows the accuracy comparison of the results obtained with the state-of-the-art is shown in table 4.2.

4.2. **Scattered plot and ROC curve**

Scatter plots are useful in statistics because they may demonstrate the amount of any association between the values of selected characteristics and the occurrences. A scatter plot's main purpose is to track and depict the relationship between two numerical variables. The scatter plot visualization of the dataset is shown in Figure 8.
The Receiver Operating Characteristics (ROC) curve for SVM, Decision tree and Random Forest is shown in figure 9. The performance of the categorization thresholds is shown by the ROC curve, The genuine positive and false positivity rates are plotted on the curve.

A heatmap is a type of data visualization that displays a phenomenon's size in two dimensions using color. The reader will receive clear visual clues about how the occurrence is clustered or fluctuates over space from the fluctuation in color, which may be by hue or intensity.

5. CONCLUSION

Although collecting a solid dataset within a small reach was tough, it was done. This work may be utilized by network engineers to aid in the enhancement of intrusion detection systems by allowing them to choose a better, quicker, more immediate, and simple method. It also allows future academics to develop and improve ways to solve the problem of intrusion detection systems, as well as protect the data of users from massive hackers.

It's crucial to note the approach to intrusion detection system development is aimed at improving it. To make this project even better, our research suggests that future work should involve expanding the model's field of use. Future researchers could also explore using real-world data as well as a larger dataset to increase the model's accuracy. Other algorithms, such as KNN, x-means might be added to increase the system's robustness, according to this study.

REFERENCES

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies. https://doi.org/10.1002/ett.4150

Almasoudy, F. H., Al-Yaseen, W. L., & Idrees, A. K. (2020). Differential Evolution Wrapper Feature Selection for Intrusion Detection System. Procedia Computer Science, 167(2019), 1230–1239. https://doi.org/10.1016/j.procs.2020.03.438

Alyousef, M. Y., & Abdelmajeed, N. T. (2019). Dynamically detecting security threats and updating a signature-based intrusion detection system's database. Procedia Computer Science, 159, 1507–1516. https://doi.org/10.1016/j.procs.2019.09.321

Ayodeji, A., Liu, Y. kuo, Chao, N., & Yang, L. qun. (2020). A new perspective towards the development of robust data-driven intrusion detection for industrial control systems. Nuclear Engineering and Technology, 52(12), 2687–2698. https://doi.org/10.1016/j.net.2020.05.012

Ayogu, B. A., Adetunmbi, A. O., & Ayogu, I. I. (2019). A Comparative Analysis of Decision Tree and Bayesian Model for Network Intrusion Detection System. FUOYE Journal of Engineering and Technology, 4(2).

https://doi.org/10.46792/fuoyejet.v4i2.362

Borkar, A., Donode, A., & Kumari, A. (2018). A survey on Intrusion Detection System (IDS) and Internal Intrusion Detection and protection system (IIDPS). Proceedings of the International Conference on Inventive Computing and Informatics, ICICI 2017, 949–953. https://doi.org/10.1109/ICICI.2017.8365277

Bul, W., James, A., & Pannu, M. (2015). Journal of Computer and System Sciences Improving network intrusion detection system performance through quality of service configuration and parallel technology. Journal of Computer and System Sciences, 81(6), 981–999. https://doi.org/10.1016/j.jcss.2014.12.012

Choudhury, S., & Bhowal, A. (2015). Comparative analysis of machine learning algorithms along with classifiers for network intrusion detection. 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings, 89–95. https://doi.org/10.1109/ICSTM.2015.7225395

Dey, A., & Learning, A. S. (2016). Machine Learning Algorithms: A Review. 7(3), 1174–1179.

Ikuomola, A, J. (2015). An evaluation of classification algorithms for intrusion detection. Journal of Computer Science and Its Application, 22(1).

Jyothsna, V., & Munivara Prasad, K. (2020). Anomaly-Based Intrusion Detection System. In Computer and Network Security. IntechOpen. https://doi.org/10.5772/intechopen.82287

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2(1). https://doi.org/10.1186/s42400-019-0038-7

Liu, H., & Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences, 9(20), 4396. https://doi.org/10.3390/app9204396

Mudzingwa, D. (2014). A study of Methodologies used in Intrusion Detection and Prevention Systems (IDPS). March 2012. https://doi.org/10.1109/SECon.2012.6197080

Neyole Misiko Jacob, B., Yusuf Wanjala, M., Misiko Jacob α, N., & Yusuf Wanjala σ, M. (2017). A Review of Intrusion Detection Systems. In Software & Data Engineering Global Journal of Computer Science and Technology (Vol. 17). C.

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020a). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. Procedia Computer Science, 171(2019), 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Khan, M. K. A. A. (2020b). Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. Procedia Computer Science, 171, 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

Sharma, R., & Kalita, H. K. (2015). Analysis of Machine Learning Techniques Based Intrusion Detection Systems. June. https://doi.org/10.1007/978-81-322-2529-4Sharma, R., & Kalita, H. K. (2015). Analysis of Machine Learning Techniques Based Intrusion Detection Systems. June. https://doi.org/10.1007/978-81-322-2529-4

Shenfield, A., Day, D., & Ayesh, A. (2018). Intelligent intrusion detection systems using artificial neural networks. ICT Express, 4(2), 95–99. https://doi.org/10.1016/j.icte.2018.04.003

Thakkar, A., & Lohiya, R. (2020). ScienceDirect A Review of the Advancement in in Intrusion Detection Datasets. Procedia Computer Science, 167(2019), 636–645. https://doi.org/10.1016/j.procs.2020.03.330

Table 1: Differences in Intrusion Detection Systems based on Host and Network

	Host-Based IDS	Network-Based IDS
Data Source	Operating Device Logs or Application Programs	Traffic on the Network
The Deployment	Every host; Operating system dependent; Hard to deploy	Main nodes for the network; Easy to deploy
Performance in detection	Low, multiple logs must be processed	High, can identify attacks in real time
Identifiability for Intrusion	Trace the intrusion mechanism based on device call paths	Detect the location and timing of an incursion using IP addresses and location information.
Restrictions	Network habits cannot be examined	Monitor only the traffic that passes via a specific network section.

Table 2: **Comparison of related works**

Authors	Methods	Result
Rai,2016	Decision trees	80.77%
Tagliaferri,2018	Decision trees and random forests	88.89%
Sarana, 2020	LDA algorithm, CART algorithm and random forest	98.1%, 98%, 99.81% respectively
Liu & Lang, 2019	LTSM and CNN	97.60%

Table 3: Evaluation Metrics for the study

Performance Measures (%)	Data + Decision Tree	DATA + C - Means + Decision Tree	Formulae
Accuracy	99.66	99.77	(TP+TN)/(P+N)
Specificity	99.69	99.65	TN/(FP+TN)
Precision	99.73	99.87	TP/(TP+FP)
Sensitivity	99.63	99.75	TP/(TP+FN)
F1-score	99.68	99.94	2TP/(2TP+FP+FN)
Negative predictive value	99.88	99.88	NPV=TN/(TN+FN)
False positive rate	0.30	0.01	FPR=FP/(FP+TN)
False discovery rate	0.34	0.06	FDR=FP/(FP+TP)
False negative rate	0.14	0.03	FNR=FN(FN+TP)
Matthew's correlation coefficient	99.55	0.04	TP*TN-FP*FN/sqrt((TP+FP) *(TP+FN) *(TN+FP) *(TN+FN))

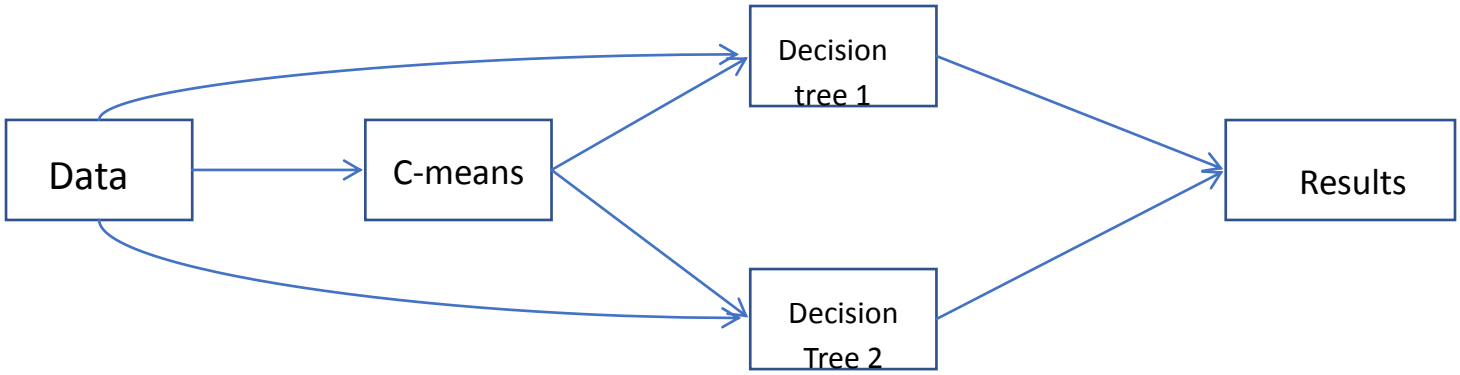

Figure 1: Proposed design

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_srv_count	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_sar
0	0	tcp	ftp_data	SF	491	0	0	0	0	0	...	25	0.17	0.03	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	0.60	
2	0	tcp	private	S0	0	0	0	0	0	0	...	26	0.10	0.05	
3	0	tcp	http	SF	232	8153	0	0	0	0	...	255	1.00	0.00	
4	0	tcp	http	SF	199	420	0	0	0	0	...	255	1.00	0.00	

Figure 2: Loaded Dataset

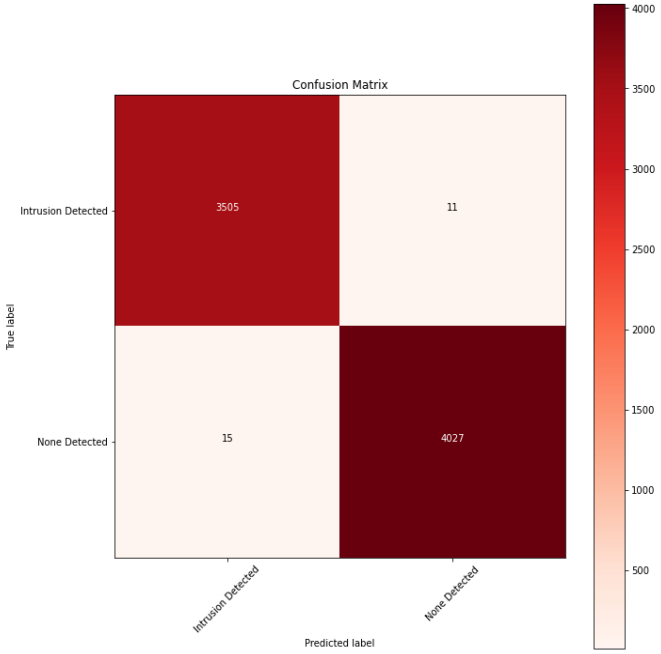

Figure 3: Confusion matrix for intrusion detection using Decision tree (TP=3505; TN=11; FP=15; FN=4027)

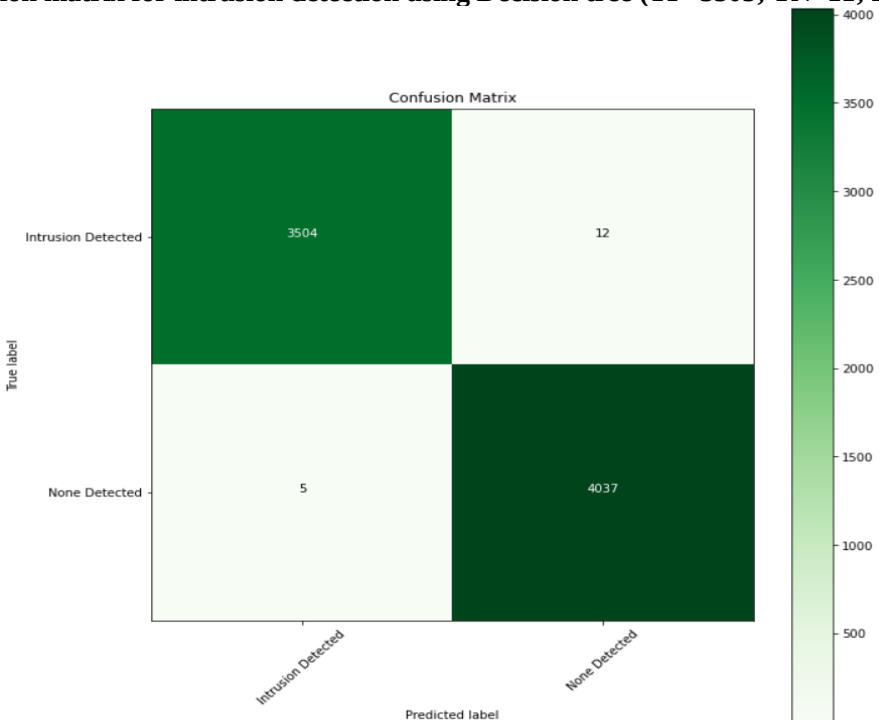

Figure 4: Confusion matrix for Intrusion Detection using Decision Tree and C-means (TP=3504; TN=12; FP=5; FN=4037)

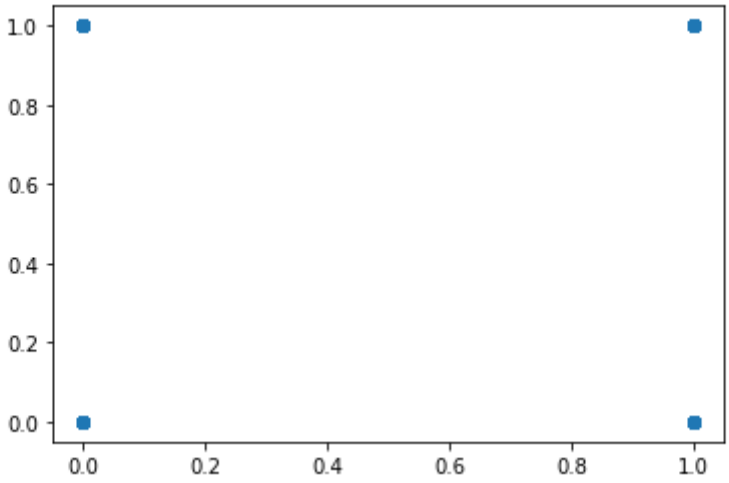

Figure 5: Scatter plot

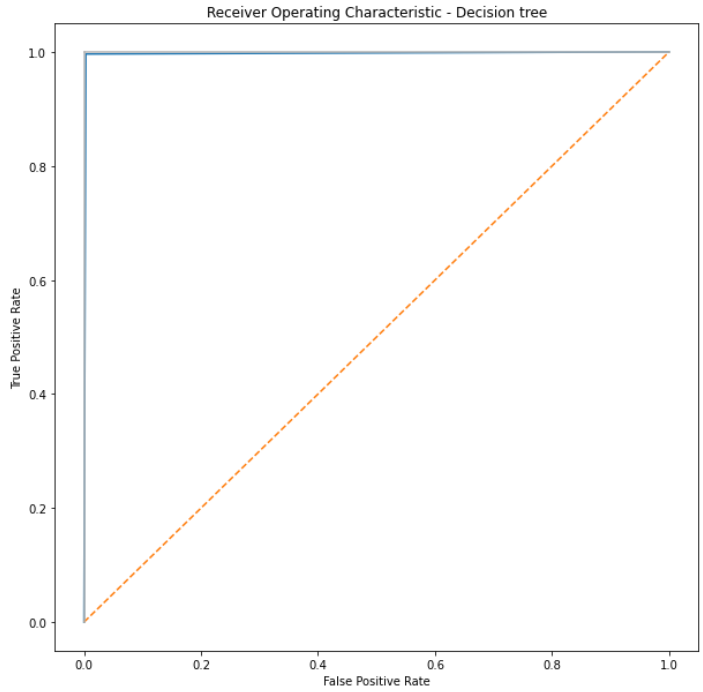

Figure 6: ROC curve of Decision tree

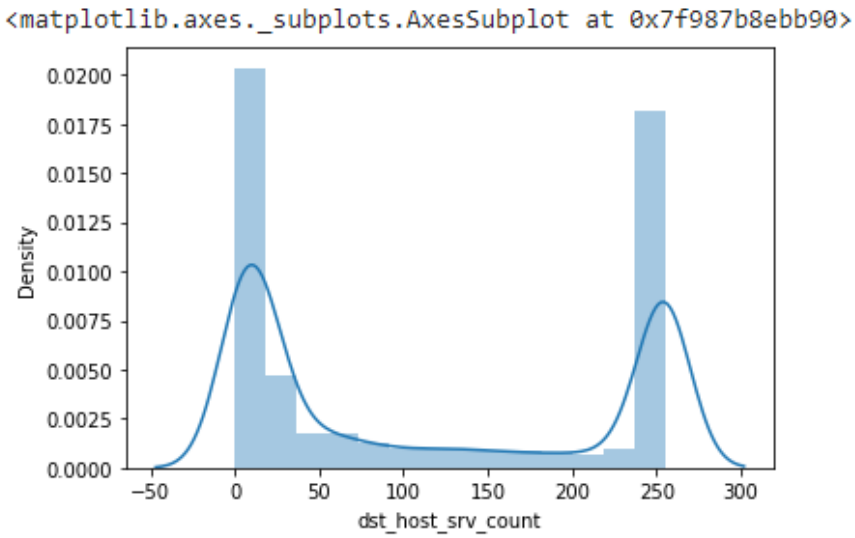

Figure 7: Dataset Subplot

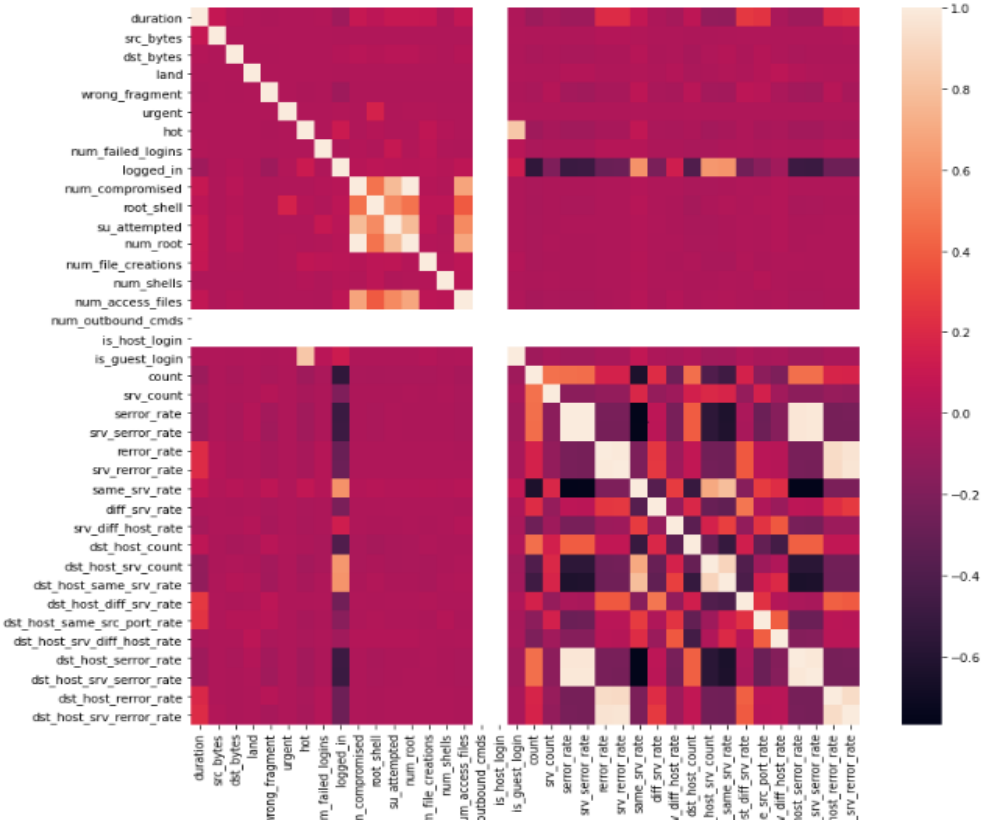

Figure 8: Heatmap for the data

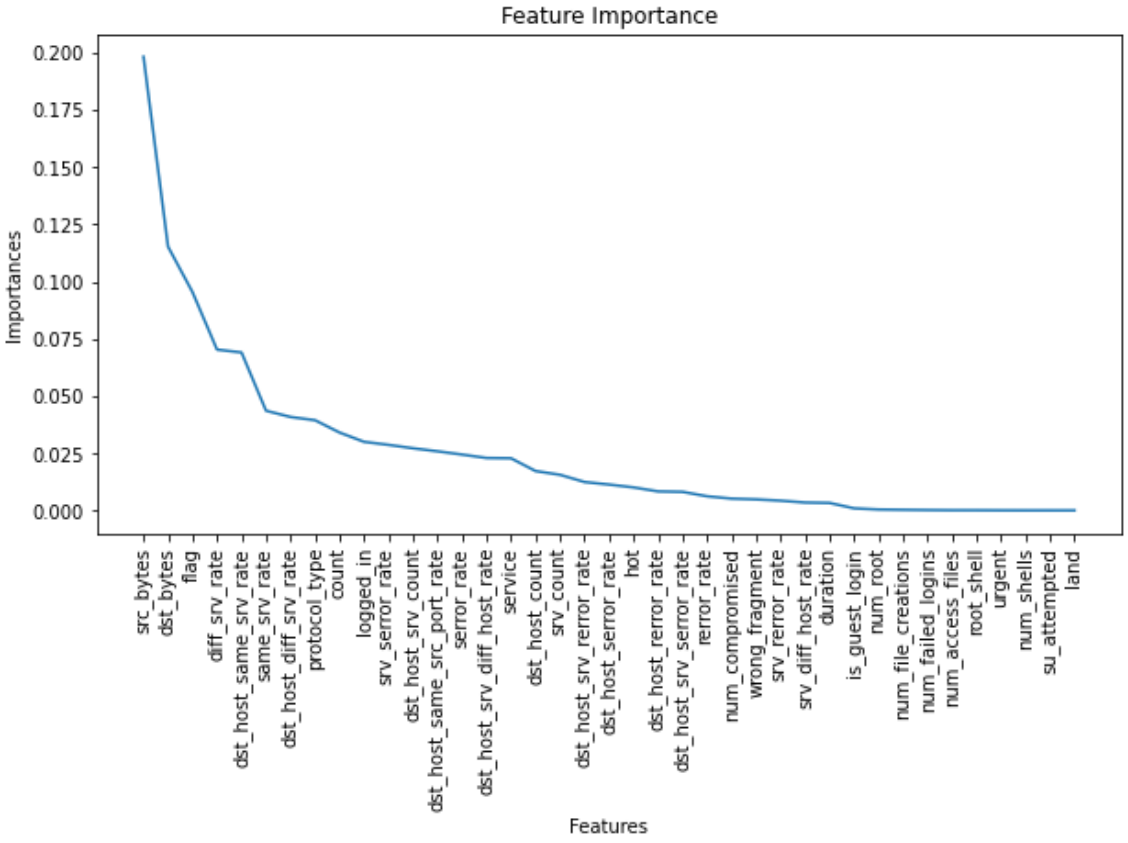

Figure 9: Data Features