



DEVELOPMENT OF AN INTRUSION DETECTION SYSTEM IN WEB APPLICATIONS USING C-MEANS AND LOGISTIC REGRESSION ALGORITHM

***DAMILOLA DAVID POPOOLA; & **DR. MICHAEL OLAOLU AROWOLO**

**Department of Computer Science, Kwara State University, Malete, Kwara State. **Department of Computer Science, Landmark University, Omu-Aran, Kwara State.*

Abstract

Intrusion detection systems have become a crucial requirement for every firm to safeguard applications from intruders. One of the security methods used as an additional line of defense to stop web security breaches is the intrusion detection system (IDS). Although it is an established tool for identifying network-based assaults, web application

security is still in its infancy. The paper's main goal is to provide a full understanding of the

Keywords: *Intrusion Detection Systems, Support Vector Machine, Logistic Regression, C-Means, K-Means*

detection system's design process in relation to online applications. It uses Support Vector Machine (SVM) and Logistic

INTRODUCTION

Intrusion detection systems (IDS) are devices or apps that monitor channels for any unwelcome actions that disrupt normal software quality and lead to policy violations. One of the most crucial security systems available today is an intrusion detection system. An IDS is a network traffic monitoring system that searches for illegal activities or network access and notifies users if these actions or access are detected (Agarwal & Hussain, 2018).

Signature-based intrusion detection systems, anomaly-based intrusion detection systems, and intrusion detection systems that mix the two are the three types of intrusion detection systems. Anomaly-based intrusion detection systems, as well as a variety of other analytical methodologies, have recently been developed and deployed to track new system threats. These techniques can achieve detection rates of 98% at a high alert rate and 1% at

Regression classifiers with a C-Means clustering on incursion datasets. All proximity metrics used in C-Means clustering approaches are included in the simulation. The confusion matrix is used to compare the accuracy of the clustering techniques. The outcome demonstrates that SVM and C-Means combine to produce superior clustering accuracy. K-Means is therefore a preferable choice for designing intelligent intrusion detection products.

a low alert rate (Agarwal & Hussain, 2018).

The study looked at the similarities and conflicts that have occurred in the development of machine learning methods and techniques for defect detection and cybersecurity in complex network defense, as well as the need to differentiate the two. Intrusion detection systems for Industrial Control Systems (ICS) are frequently trained on network packet captures and focus primarily on monitoring network layer traffic for intrusion detection. The intrusion detection system employs machine learning techniques to identify and recognize security threats (Dini & Saponara, 2021).

For a range of applications, such as fog computing, the Internet of Things (IoT), big data, smart cities, and 5G networks, intrusion detection systems apply machine learning approaches. Machine learning algorithms that can be used to detect infiltration include Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest (Shrestha et al., 2021).

Failure to recognize and differentiate 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 false alarm rate for nuclear power plant control systems that account for observable process calculation shifts and intrusion detection systems used to identify intrusions on industrial controllers is significant (Park et al., 2020).

To target computer users, cybercriminals deploy cutting-edge tactics and social engineering techniques. With the passage of time, fraudsters get more sophisticated and inspired. Cybercriminals have demonstrated their abilities to conceal their identities, obscure their messages, keep their identities separate from illicit earnings, and use secure technologies. As a result, advanced intrusion detection solutions capable of identifying current ransomware are becoming more critical for the security of computer systems. To plan and construct such Intrusion Detection System programs, a thorough understanding of the current Intrusion Detection System research is required (Alkhalil et al., 2021).

This paper suggests the use of a machine learning approach to detect intrusion in online applications by classifying and reviewing machine learning-based approaches for cyber security experts.

Literature review

Mudzingwa, (2014) used four main detection methodologies used by IDPS are signature based, anomaly based, stateful protocol analysis based, and hybrid based. While the anomaly-based methodology outperforms the other two in terms of identifying new threats without requiring any user feedback or modifications, the majority of users prefer the other two approaches. A variation of the four major methodologies is used in existing IDPS on the market. It also made it simple to evaluate and test the IDPS approaches utilized by current IDPS products.. Experiments with commercial and open source software, as well as our assessment criteria, will be part of its future study.

Shenfield (2018), proposed artificial neural networks were offered as a unique way to detecting malicious network traffic that might be used in deep packet inspection-based intrusion detection systems. The proposed artificial neural network architecture can be destituted using 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. Through repeated 10-fold cross-validation, the proposed artificial neural network architecture 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.

Sharma, (2015) applied machine learning methods which has been shown to be effective at detecting intrusions. Machine learning techniques can achieve high accuracy in intrusion detection, but the detection accuracy is often dependent on other factors. Selection of the appropriate feature set, training and testing data, and so on are just a few of them. By choosing the proper qualities for these variables, you can get a better result. On the other hand, machine learning algorithms may contain defects, such as misrepresentation of network data owing to poison learning.

Dey , (2016) also used various machine learning algorithms used for various purposes like data mining, image processing, predictive analytics The main advantage of machine learning is that once an algorithm understands how to deal with data, it can execute tasks on its own.

Thakkar, (2020) examines datasets created in the field of Intrusion Detection Systems (IDS). These datasets were used to evaluate the effectiveness of IDS based on machine learning and data mining. According to the findings, the underlying dataset should be updated in order to better distinguish newer assaults in the field of IDS. This is due to the fact that attackers employ a wide range of processes and technology in their attacks. Furthermore, the process of launching different attacks replicates the necessity for datasets with plausible network conditions. They'll focus on assessing the output of these datasets using a range of machine learning and data mining approaches in the future, as well as incorporating feature engineering and data sampling to fix the defects in the datasets.

Ayogu, (2019), proposed the collection of important qualities was used as a decision tree, which has the benefit of generating understandable rules. A C4.5 decision tree dimensional reduction approach was employed to reduce the 41 features of the KDD'99 dataset to 29. Then, based on the selected attributes, a rule-based classification system (decision tree) was created, as well as a Bayesian denial of service attack (DoS) network classification system. The classifiers were assessed and compared using the findings on the test dataset. The Decision Tree, according to experimental data, is more stable and provides the highest percentage of effective classification than the Bayesian Network, which has been found to be susceptible to discretization approaches. Significant attribute selection has been proven to be crucial in constructing a real-world intrusion detection system (IDS).

Ahmad (2020), worked on a systematic approach to machine learning, explaining the IDS definition and then presenting a taxonomy based on prominent ML and DL approaches used in the construction of network-based IDS (NIDS) systems. A complete overview of current NIDS-based articles is offered by examining the strengths and weaknesses of the proposed solutions. Then, recent trends and advancements of ML and DL-based NIDS are discussed in terms of recommended methodology, assessment metrics, and dataset collecting. Using the weaknesses of the presented approaches, we presented a number of research concerns and the potential scope for study to create ML and DL-based NIDS. Ayodeji, (2020) recent improvements in intrusion detection algorithms were investigated, as well as their flaws, limitations, and current position in critical infrastructure applications. We also look at the parallels and conflicts that have been discovered in the use of machine learning tools and techniques for fault detection and cybersecurity in complex system protection, as well as the necessity to differentiate between the two. In this study, a major contributor to false alarm generation is the 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. To support his position, he looked at the 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.

Bul, (2015) described an advanced software development that uses Cisco Catalyst Switches' Quality of Service (QoS) and parallel techniques to enhance the analytical performance of a Network Intrusion Detection and Protection System (NIDPS) when deployed in high-speed networks, and designed a real network to present experiments using a Snort Network Intrusion Detection and Protection System. An Intrusion Detection and Protection System, on the other hand, is widely regarded as one of the most effective technologies for detecting threats and assaults. Many organizations and governments have shown interest in Network Intrusion Detection and Protection Systems, and anyone with access to the Internet can use them. Scanning, analysing, detecting, and correcting are the four stages of a Network Intrusion Detection and

Protection System for securing a computer system network. The emphasis of our paper was on Network Intrusion Detection and Protection System' scanning and analysis flaws in high-speed network connectivity. To boost NIDPS analysis efficiency and reduce Network Intrusion Detection and Protection System processing time, we recommend using a QoS configuration and parallel technology. As a consequence of our approach, devices can be designed to make it easier to thwart attacks. The present and expected potential demands for online security necessitate the revision of existing systems in order to create better parallel systems and more robust rule sets.

Almasoudy, (2020) proposed Different evolution technique for intrusion detection system. The main idea is to utilize Differential Evolution to select some features from the NSL-KDD datasets' 41 features, and then utilize Extreme Learning Machine to compute the accuracy of these features. Differential Evolution is carried out until the fewest number of high-accuracy features is attained. In both five and binary classification, the results showed a higher detection rate and a lower false alarm rate. The proposed system attained an accuracy of 80.15 percent for five classification and 87.53 percent for binary classification with a reduction in training and testing time. In the future, we intend to develop links from live networks. 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.

Veeramreddy and Munivara (2020), worked on anomaly-based intrusion detection system. A crucial challenge is that it is independent of the research in a fast and sufficient pre-processing phase. Another difficult problem is the increasing increase of zero-day attacks, which highlights the necessity for security systems that can precisely detect previously unknown attacks. Through the use of active feature optimization approaches, an attempt is made to develop a generic meta-heuristic scale for both known and undiscovered assaults with a high detection rate and low false alarm rate.

Akinyemi (2019), improved identification accuracy by eliminating the imbalanced class issue that is frequently associated with network traffic datasets live network traffic packets were gathered during routine network operations, Sync flood assault, sluggish http post attack, and exploitation of known vulnerabilities on a targeted computer in the test case setting with Wire shark. The Spleen program was used to extract 52 features from the packet meta-data, including 42 features that were identical to the Information Discovery in Database (KDD'99) intrusion detection dataset. The characteristics were normalized using the min-max normalization algorithm, and the Knowledge Gain algorithm was used to select the best discriminatory features from the feature space. An anomalous intrusion detection model formulated a cascade of k-means clustering algorithm and random-forest classifier. In comparison to the present model, the evaluation result indicated a 10% increase in detection accuracy, a 29% increase in sensitivity, and a 0.2 percent increase in specificity.

Ikuomola (2015), suggested that principal Component Analysis (PCA) was used to reduce the features in the dataset in order to lower the amount of computer resources required to detect attacks, and the experiment was conducted using the KDD cup 99 dataset. The results demonstrate that the Neural Network algorithm (Nnge) performed better than the other systems since it was able to create a minimum false positive rate of 0.9 percent and 1% before and after the function reduction. The false negative alert rate was 3.2 percent before and after feature removal, with the maximum classification accuracy.

Liu and Lang (2019), worked on a machine learning methodology. The IDS taxonomy was created to identify machine learning-based and deep data objects, with the key dimension being to summarize them. IDS literature that is based on learning. This type of classification structure, we believe, is appropriate for cyber security researchers. Second, the survey clarifies the definition and taxonomy of IDSs. After that, the most commonly used machine learning methods are implemented in IDSs, metrics, and benchmark datasets. Then, using the suggested taxonomic structure as a foundation, we show how to solve important IDS concerns using machine learning and deep learning approaches, along with examples from the literature. Finally, concerns and prospective trends are examined by reviewing recent representative reports.

Mahani and Ali (2020), worked on machine learning techniques for intrusion detection were created as a result of the effort. Network security is one of the most urgent issues in today's world. Network security vulnerabilities have become a serious concern as the Internet has grown exponentially and been more widely used over the previous decade. Over secured networks, an intrusion detection device is used to detect unauthorized access and unexpected attacks. In recent years, several studies on intrusion detection systems have been conducted. However, to better understand the current status of machine learning, this survey research analyzed 49 related works from 2009 to 2014 that focused on single, hybrid, and ensemble classifier design architecture.

Ayo, (2020) suggested that to promote innovation and comparison, feature selection, numerous search algorithms, and attribute evaluators have been merged. The findings revealed that the number of features chosen had no effect on the detection accuracy of function selection algorithms, but was directly related to the output of the base classifier. The proposed methodology surpasses other similar methods in terms of false alarm rate, high accuracy rate, decreased training and testing time, 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. Additionally, simulation studies using common assessment metrics revealed that the suggested technique is suitable for NIDS attack classification.

Method and Materials

In this study, the system consists of developmental stages: the pre-processing and Classification. The dataset loads as an input; it is pre-processed using C-means

clustering, it obtains the relevant information in the dataset, the features are classified into clustering and the results are compared.

The C-Means clustering approach is utilized to partition the dataset in this study; it is an unsupervised learning process for solving clusters with the nearest mean, which acts as a prototype for the cluster. The classification methodology is based on a decision tree. The classification performance is used to produce the findings for this methodology. The following is a breakdown of the methods used in this study:

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

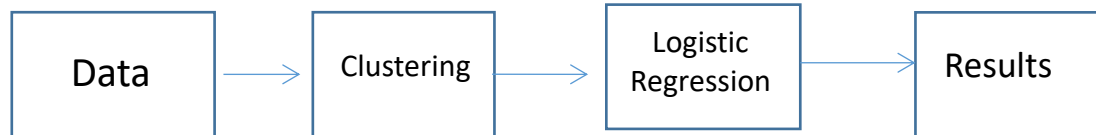


Figure 1: Proposed design

Step 1- Using Clustering algorithm method; pre-process the churn customer dataset.

Step 2- Using Logistic Regression, as a Classification algorithm to carry out and simplify the performance of the dataset, so as to predict and improve intrusion level.

Experimental Dataset

To accomplish the intended result, the suggested system uses an intrusion detection dataset from CIC-IDS2017 (Friday-WorkingHours-Afternoon-DDos.pcap ISCX). The dataset contains harmless and current common attacks, and it closely reflects real-world data (PCAPs). It also includes the findings of a CICFlowMeter network traffic analysis, which includes labeled flows based on the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack vectors (CSV files).

Clustering Algorithm

C-means will be used for the clustering techniques in this study to to examine the difference of efficiency performance.

C-means clustering is an unsupervised learning technique that is used when there is unlabelled data, or data that does not have specified categories or groups. The number of groups in a data set is expressed by the variable C, and the algorithm finds them. Based on the features given, this algorithm assigns each data point to one of C groups iteratively. Although the C-means clustering algorithm works well with compact and hyper-spherical clusters, we're interested in pointing out its flaws and proposing solutions. The emphasis is primarily on two unavoidable C-means algorithm problems:

- (i) the number of clusters and centroids assigned, and
- (ii) the ability to manage various types of data (Mohammed, 2020).

Pseudo code for C-means clustering

Function Direct-C-means()

initialize k prototypes ($w_1, \dots, w_k$) such that $w_j =$

$i_j, j \in \{1, \dots, k\}, i \in \{1, \dots, n\}$

each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_j , where $i \in \{1, \dots, n\}$,

do

Assign i_j to the cluster C_{j^*} with nearest prototype w_{j^*}

(i.e, $|i_j - w_{j^*}| \leq |i_j - w_j|, j \in \{1, \dots, k\}$)

For each cluster C_j , where $j \in \{1, \dots, k\}$,

do

Update the prototype w_j to be the centroid of all samples currently in C_j ,

So that $w_j = \sum_{i_j \in C_j} i_j / |C_j|$

Compute the error function :

$E = \sum_{i_j \in C_j} |i_j - w_j|^2$

Until E does not change significantly or cluster membership no longer changes

Classification Algorithm

A Decision Tree is a tree-like graph with internal nodes representing a test on an attribute, branches representing the test's result, and leaf nodes representing the class label. The classification rules are determined by the path taken from the root node to the leaf. The root node is picked first to split each input data set because it is the most noticeable property. The tree is formed by describing features and their associated values at each intermediary node, which will be utilized to evaluate the input data. It's capable of analyzing data and detecting key network features that point to malicious behavior. It can bring value to many real-time security systems by analyzing a big quantity of intrusion detection data. It can detect patterns and trends that can aid in further research, the expansion of attack signatures, and other monitoring duties. The main advantage of decision trees over other classification approaches is that they provide a comprehensive set of rules that are easy to understand and integrate with real-time systems. (Rai, 2016).

Pseudo code for Decision Tree

Procedure DT inducer (S,A,y)

1. Create a tree T
2. if one of the stopping Criteria is fulfilled then
3. Mark the root node in T as a leaf with the most common value of y in S as the class
4. Else
5. Find a discrete function $f(A)$ of the input attributes values such that splitting S according to $f(A)$'s outcome

6. if best splitting metric $\geq$ threshold then
7. label the root node in T as $f(A)$
8. for each outcome v_i of $f(A)$ do
9. subtree_i = TreeGrowing ($O_{f(A)=v_i} S, A, y$).
10. connect the root node of T to subtree_i with an edge that is labelled as v_i
11. end for
12. else
13. Mark the root node in T as a leaf with the most common value of y in S as the class.
14. end if
15. end if
16. Return T

Procedure tree pruning (S, T, y)

1. Repeat
2. Select a node t in T such that pruning it maximally improve some evaluation criteria
3. if $t \neq \emptyset$ then
4. T = pruned (T,t)
5. end if
6. until $t = \emptyset$
7. Return T

Performance Metrics

A. Sensitivity

The percentage of real positives that are properly identified as positives is measured by sensitivity.

$$\text{Sensitivity} = \text{TP} / (\text{TP} + \text{FN}) \%$$

B. Specificity

The fraction of actual negatives that are accurately classified as negatives is measured by specificity, also known as selectivity or true negative rate (TNR).

$$\text{Specificity} = \text{TN} / (\text{TN} + \text{FN}) \%$$

C. Accuracy

The number of correct predictions made by the model out of all sorts of predictions generated is characterized as accuracy in classification problems. Accuracy is a good measure when the target variable classes in the data are roughly balanced.

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN}) \%$$

D. Precision

Precision is a measure that indicates how accurate our estimates were on a percentage basis.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

Where:

TN (True Negative) = correctly classified negative cases,

TP (True Positives) = correctly classified positive cases,
 FN (False Negative) = incorrectly classified positive cases.
 FP (False Positives) = incorrectly classified negative cases,
 Sensitivity (true positive fraction) is the likelihood that a diagnostic test will be positive if the person has the condition.
 Specificity (true negative fraction) is the probability that a diagnostic test will be negative if the person does not have the condition.
 Accuracy is the possibility that a diagnostic test will be done accurately

Result and Discussions

The raw data is sent straight to the classifiers SVC and Logistic Regression in the first phase, and the raw data will be 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 25192 characteristics and 42 attributes, are then compared. The jupyter environment used to implement the project is shown in Figure 4.1.

```
Training data has 25192 rows & 42 columns
```

	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_same_srv_per
0	0	tcp	ftp_data	SF	401	0	0	0	0	0	...	20	0.07	0.00	
1	0	udp	other	SF	146	0	0	0	0	0	...	1	0.00	0.00	
2	0	tcp	private	SF	0	0	0	0	0	0	...	20	0.00	0.00	
3	0	tcp	http	SF	230	3153	0	0	0	0	...	230	1.00	0.00	
4	0	tcp	http	SF	166	400	0	0	0	0	...	330	1.00	0.00	

```
5 rows x 42 columns
```

Figure 2: Loaded Data

The data collection is cleaned using the C-means clustering approach in order to make it usable. In machine learning, this is often used to divide 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, figure 4.2 shows the plot for the data

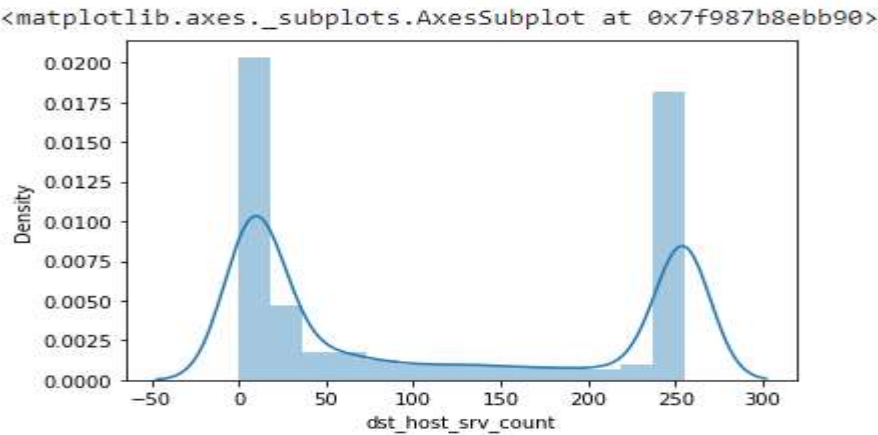


Figure 3: Data Plots

In this study, data is classified using C-Means with SVC and Logistic Regression; nevertheless, the data is fed into the specified classifiers, thus C-means clustered data is also passed into the classifiers, as seen in figure 4.3, figure 4.4, figure 4.5, and figure 4.6.

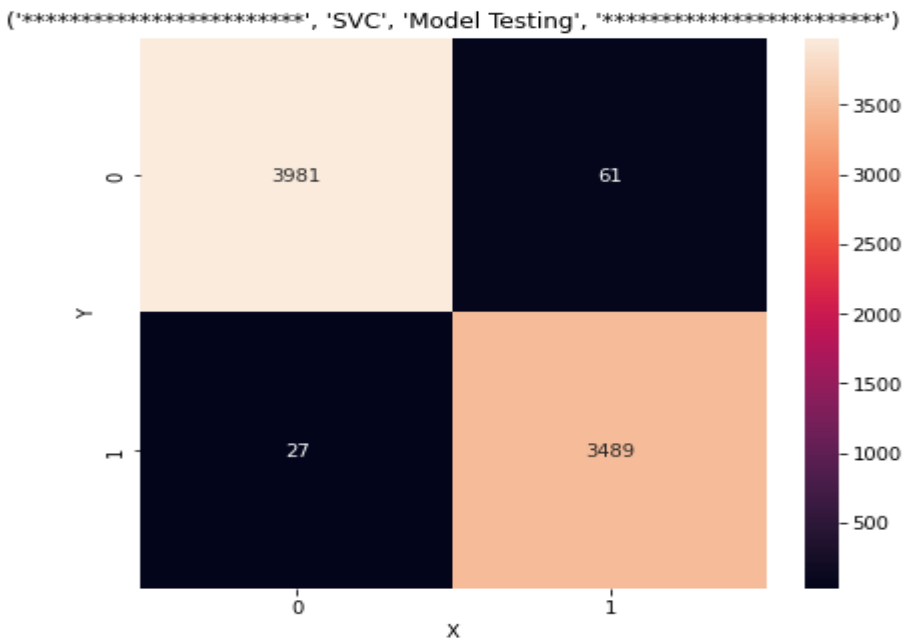


Figure 4: C-Means with SVM (TP= 3981; TN =3489; FP=61; FN=27)

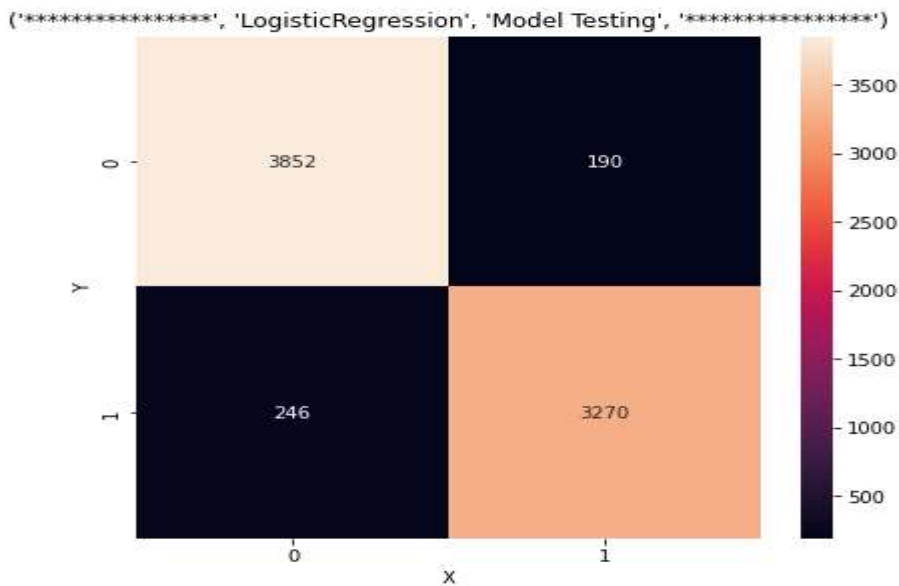


Figure 5: Confusion Matrix for C-Means with Logistic Regression (TP=3852; TN=3270; FP=190; FN=246)

The sensitivity, specificity, precision, negative predictive value, false positive rate, false discovery rate, false negative rate, accuracy, F1 score, and Matthew's correlation coefficient are used to assess the confusion matrices generated. The experiment

assessment measures are shown in Table 4.1.

Table 1: Result Evaluation

Measure	C-Means + SVM	C-Means +LR	Derivations
<u>Sensitivity</u>	99.33	95.30	$TPR = TP / (TP + FN)$
<u>Specificity</u>	98.28	93.00	$SPC = TN / (FP + TN)$
<u>Precision</u>	98.49	5.49	$PPV = TP / (TP + FP)$
<u>Negative Predictive Value</u>	99.23	4.70	$NPV = TN / (TN + FN)$
<u>False Positive Rate</u>	1.72	6.00	$FPR = FP / (FP + TN)$
<u>False Discovery Rate</u>	1.51	94.23	$FDR = FP / (FP + TP)$
<u>False Negative Rate</u>	0.67	94.64	$FNR = FN / (FN + TP)$
<u>Accuracy</u>	98.84	88.40	$ACC = (TP + TN) / (P + N)$
<u>F1 Score</u>	98.91	95.30	$F1 = 2TP / (2TP + FP + FN)$
<u>Matthews Correlation Coefficient</u>	97.67	93.00	$TP*TN - FP*FN / \text{sqrt}((TP+FP)*(TP+FN)*(TN+FP)*(TN+FN))$

Several tests were conducted in this investigation, and the results are shown in table 4.1; nevertheless, c-means + decision tree outperformed with 99% accuracy. The accuracy comparison of the result achieved with the state-of-the-art is shown in Table 4.2.

Table 1: 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%

Conclusion

This is a huge project that will help network engineers identify intrusions. 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 any online application from massive hackers.

Recommendations

It's crucial to note that 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 Radom forest, X-means, and fuzzy logic, might be added to increase the system's robustness, according to this study.

REFERENCE

- Agarwal, N., & Hussain, S. Z. (2018). A Closer Look at Intrusion Detection System for Web Applications. *Security and Communication Networks*, 2018, 1–27. <https://doi.org/10.1155/2018/9601357>
- 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>
- Alkhalil, Z., Hewage, C., Nawaf, L., & Khan, I. (2021). Phishing Attacks: A Recent Comprehensive Study and a New Anatomy. *Frontiers in Computer Science*, 3. <https://doi.org/10.3389/fcomp.2021.563060>
- 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>
- 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/fuoyej.v4i2.362>
- 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>
- Dey, A., & Learning, A. S. (2016). *Machine Learning Algorithms : A Review*. 7(3), 1174–1179.
- Dini, P., & Saponara, S. (2021). Analysis, Design, and Comparison of Machine-Learning Techniques for Networking Intrusion Detection. *Designs*, 5(1), 9. <https://doi.org/10.3390/designs5010009>
- Ikuomola, A. J. (2015). An evaluation of classification algorithms for intrusion detection. *Journal of Computer Science and Its Application*, 22(1).
- Jang-Jaccard, J., & Nepal, S. (2014). A survey of emerging threats in cybersecurity. *Journal of Computer and System Sciences*, 80(5), 973–993. <https://doi.org/10.1016/j.jcss.2014.02.005>
- Jose, S., Malathi, D., Reddy, B., & Jayaseeli, D. (2018). A Survey on Anomaly Based Host Intrusion Detection System. *Journal of Physics: Conference Series*, 1000, 012049. <https://doi.org/10.1088/1742-6596/1000/1/012049>
- 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., & Alazab, A. (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, 4(1), 18. <https://doi.org/10.1186/s42400-021-00077-7>
- 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>
- Park, Y.-J., Fan, S.-K. S., & Hsu, C.-Y. (2020). A Review on Fault Detection and Process Diagnostics in Industrial Processes. *Processes*, 8(9), 1123. <https://doi.org/10.3390/pr8091123>
- Sarker, I. H. (2021). Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Computer Science*, 2(3), 160. <https://doi.org/10.1007/s42979-021-00592-x>
- Sharma, R., & Kalita, H. K. (2015). Analysis of Machine Learning Techniques Based Intrusion Detection Systems. June. [https://doi.org/10.1007/978-81-322-2529-4](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., & Ayes, A. (2018). Intelligent intrusion detection systems using artificial neural

- networks. *ICT Express*, 4(2), 95–99. <https://doi.org/10.1016/j.ict.2018.04.003>
- Shrestha, R., Omidkar, A., Roudi, S. A., Abbas, R., & Kim, S. (2021). Machine-Learning-Enabled Intrusion Detection System for Cellular Connected UAV Networks. *Electronics*, 10(13), 1549. <https://doi.org/10.3390/electronics10131549>
- Thakkar, A., & Lohiya, R. (2020). ScienceDirect A Review Review of of the the Advancement Advancement in in Intrusion Intrusion Detection Detection Datasets Datasets. *Procedia Computer Science*, 167(2019), 636–645. <https://doi.org/10.1016/j.procs.2020.03.330>
- Yaacoub, J.-P. A., Noura, H. N., Salman, O., & Chehab, A. (2022). Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, 21(1), 115–158. <https://doi.org/10.1007/s10207-021-00545-8>