Proceeding of 2024 International Conference on Computing and Intelligent Reality Technologies (ICCIRT)
IEEE Xplore Part Number: CFP24OZ6-ART; ISBN: 979-8-3315-1029-9

Smart Signature-based Intrusion Analysis System with Pre-Scaled Learning for Large Networks

Hemanth Sai Vardhan Balivada
Amrita School of Computing,
Coimbatore, India
cb.en.u4cse22464@cb.students.amrita.edu

Shanmuka Vardhan Bandarupalli
Amrita School of Computing,
Coimbatore, India
cb.en.u4cse22461@cb.students.amrita.edu

Dasari Vishal
Amrita School of Computing,
Coimbatore, India
cb.en.u4cse22416@cb.students.amrita.edu

Soma Siva Pravallika,
Amrita School of Computing, Coimbatore, India
cb.en.u4cse22440@cb.students.amrita.edu

Manas Pandey
Software Architect. IBM
Bangalore, India
manas.pandey@in.ibm.com

Senthil Kumar Thangavel
Amrita School of Computing,
Coimbatore, India
t_senthilkumar@cb.amrita.edu

Somasundaram K
Department of Mathematics
Amrita School of Physical Sciences
Amrita Vishwa Vidyapeetham
Coimbatore, Tamil Nadu, India
s_sundaram@cb.amrita.edu

2024 International Conference on Computing and Intelligent Reality Technologies (ICCIRT) | 979-8-3315-1029-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCIRT59484.2024.10921939

Abstract— [Intrusion detection systems are crucial elements of contemporary cybersecurity methods. They identify unauthorized actions that jeopardize integrity, confidentiality, and availability of information systems. This work extends analysis of Intrusion Detection Systems (IDS), comparing Signature-based Intrusion Detection Systems (SIDS) and Anomaly-based Intrusion Detection Systems (AIDS) in the context of threat model recognition. It further examines the integration of machine learning into an Intrusion Detection System (IDS) to enhance the detection of unknown threats, anomaly identification, and scalability. The study compares several clustering algorithms, specifically K-Means and DBSCAN, and presents a comparative analysis that demonstrates K-Means is more effective in differentiating benign from malicious data. It encompasses the installation of Intrusion Detection Systems (IDS) across several platforms, including cloud and host-based systems, and outlines approaches such as monitoring data utilization, registry modifications, and analyzing TCP dumps for intrusion detection. In the end, it delineates future directions, emphasizing advancements in AI and machine learning, quantum resistance, and privacy-preserving techniques to address the escalating security threats.]

Keywords- Intrusion Detection Systems, machine learning, K-Means, network security, cloud-based IDS, TCP dumps, registry edits)

I. INTRODUCTION

Intrusion denotes any unauthorized action that causes damage to an information system. Intrusion can manifest in different forms, including physical intrusion and digital infiltration. In today's environment, where individuals lacking a digital imprint are deemed nonexistent, digital infiltration can inflict significant suffering and distress on individuals. In contemporary society, all tasks, whether little or significant, such as ticket booking, bill payment, and medical appointments, are conducted through digital platforms. Therefore, it is crucial and advantageous to facilitate the seamless detection of instructions in the digital realm.

The broad three categories are:

(i) A **masquerader** is a someone who attempts to unlawfully steal data.

(ii) **Misfeasor** - Refers to an individual who has authorized access to data but intentionally misuses it and assists a masquerader.

(iii) **Clandestine** users assert dominance over the system and restrict access for authorized users.

Intrusion can result in various adverse consequences, such as:

- Privacy Loss: Intrusion refers to the unauthorized entry into your personal area, causing feelings of insecurity and vulnerability.
- Faith Issues: When an individual you have placed your faith in violates your privacy, it can harm your relationship and create difficulties in reestablishing trust.
- Loss of Control: Intrusion can induce a sense of relinquishing control over one's own life and surroundings.
- Security Risks: Particularly in the digital realm, can result in the unauthorized access to personal information, resulting in identity theft, financial harm, and various other security concerns.

The consequences of infiltration can be highly impactful, as it enables the theft of personal and organizational information, resulting in financial losses. The primary objective of an Intrusion Detection System (IDS) is to promptly detect various forms of data exfiltration, a task that

979-8-3315-1029-9/24/$31.00 ©2024 IEEE

cannot be accomplished by a conventional firewall. Intrusion Detection Systems (IDS) utilize a Switched Port Analyzer or Text Access Port Analyzer to examine a duplicate of the inline data in network traffic, so avoiding any interference with the actual communication. Therefore, IDS will not have any impact on the functioning of the system. They fail to prevent threats from entering the network.

The Intrusion Detection System (IDS) offers two forms of protection.

(i)**Active**: It employs active measures by promptly responding to any suspicious activity. In such cases, it logs off the user and reconfigures the firewall to prevent any further traffic from the malicious source.

(ii)**Passive**: It utilizes passive methods to identify potential security breaches. When detected, it secures the information and sends a signal to the administrator.

Intrusion Detection Systems (IDS) can be classified into two main categories: Signature-based IDS (SIDS) and Anomaly-based IDS (AIDS).

(i) **SIDS** - Utilize network traffic and logged data to detect potential known attacks by comparing them to current attack patterns. It detects attacks by exclusively examining specified patterns. It can readily detect attacks that have been previously recognized and possess a signature. It is unable to detect novel or unfamiliar attacks due to its lack of knowledge on the patterns associated with fresh ones. This method employs automated threat detection, which is more efficient than human searches.[4]

(ii) **AIDS** – developed to mitigate unforeseen assault patterns. This approach continuously monitors the activities of the system. It categorizes behavior as either normal or anomalous. Instead, then depending on its own attack patterns, it utilizes a predetermined set of rules to recognize anomalies.[4]

Benefit: One advantage is that it can detect atypical patterns since it does not depend on pre-existing signatures. It detects anomalous behaviors without precise knowledge of their underlying causes. Unlike signature-based systems, which may require specific configurations, this method functions effectively across several operating systems.

The outline is as below:

Section II examines the pertinent literature concerning intrusion detection.

Section III discusses the technique and dataset employed in this investigation.

Section IV examines the outcomes obtained from the application of clustering methods to the dataset.

Section V explains the conclusions obtained after applying algorithms on the dataset.

Section VI delineates the prospective body of work that may be undertaken.

II. RELATED WORKS

The various techniques on Android malware, their corresponding incapacitations, and the paper on zero-day malware and obfuscated threats are discussed. It finds that most of the existing techniques have failed when sophisticated attacks are launched against the Android-based systems because of the very popularity of Android and vulnerabilities found in the systems. Hence, improved detection techniques gain emphasis in this study[1]. This paper reviews the use of Artificial Intelligence (AI) in malware detection, which focuses on shallow and deep learning models against evolving threats. An important aspect of this area has been the challenges: data acquisition proves difficult; findings cannot be compared from one study to another. The paper classifies malware detection techniques in depth and provides guidelines for novice researchers to understand the research flow and identify future directions of AI-based malware detection[2]. IDS refers to the monitoring process of network traffic or devices regarding suspicious or malicious activities, primarily through signature-based and anomaly-based detection. It is said to alert security teams to potential threats but does not prevent attacks alone; more often, it is combined with Intrusion Prevention Systems (IPS). IDS can be network-based (NIDS) or host-based (HIDS), which gives broader monitoring or endpoint-specific monitoring[3]. The paper reviews various techniques for intrusion detection through approaches based on machine learning: comparison of single, hybrid, and ensemble classifiers over seven datasets. Challenges in anomaly detection are pointed to, and a path is provided for future IDS research[5-6].

A Network Based Intrusion Detection System (NIDS) deploys sensors within a demilitarized zone (DMZ), which serves as a subnetwork safeguarding an organization's internal network from untrusted traffic [3]. It has the capability to monitor each individual data packet that enters and exits. Multiple instances of Intrusion Detection Systems (MIDS) may be required depending on the volume of network traffic. Host Based Intrusion Detection System (HIDS) is a security tool that focuses on examining and evaluating activities such as changes made to the file system, registry, or access control list [3]. Cloud-based Intrusion Detection System (CIDS) requires the deployment of network-based sensors within a cloud environment. Cloud-based servers utilize specialized cloud sensors equipped with CSP (content security policy) applications.

III. METHODOLOGY

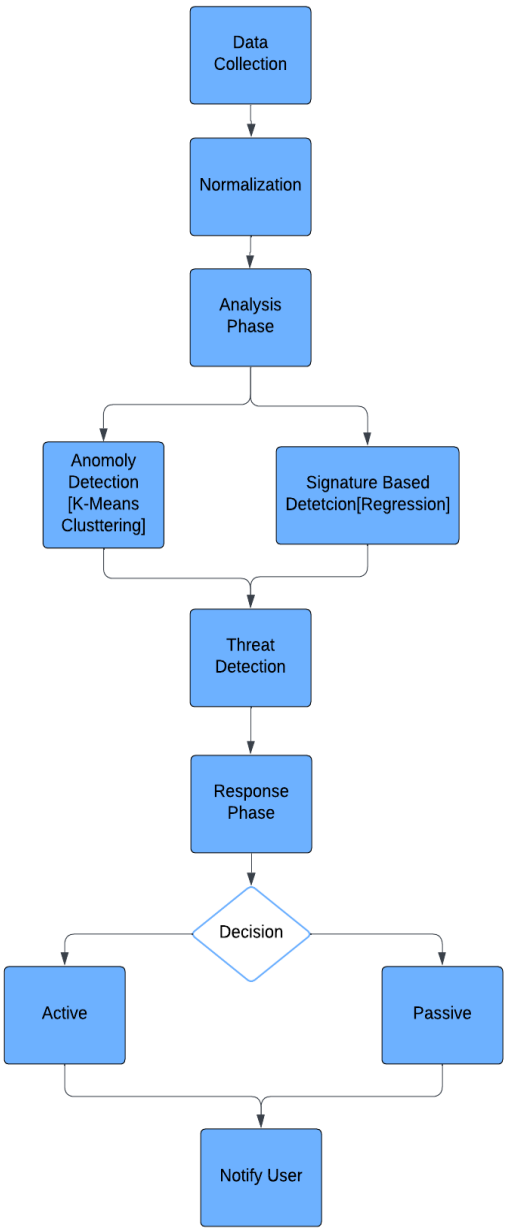
Dataset Description

DATASET:
https://www.kaggle.com/datasets/ishasingh03/friday-workinghours-afternoon-ddos

The dataset "Friday-working Hours" contains 85 columns and 9 lakh rows. It includes features such as Flow Duration, Total Fwd Packets, Total Backward Packets, Destination Port, and a Label column that can indicate either BENIGN or indicative of DDoS (Distributed Denial of Service) attacks. It also provides identifiers like Flow ID and IP-oriented information (Source IP (SI), Destination IP). The dataset is essential for building and testing an IDS with a

possibility for anomaly detection and pattern analysis for legitimate and malicious traffic.

To identify data leakage from applications, we initially observe the quantity of data transmitted by each application over a period. Through the analysis of present data usage in relation to past patterns, we can identify any atypical surges. If an application experiences a rapid increase in data consumption beyond its normal usage, it may suggest the occurrence of a data breach or prohibited data transfer.



• **Registry edits** refer to the process of monitoring and modifying the system registry, which is responsible for storing crucial settings and configurations for the operating system. By monitoring these alterations, we can detect any unlawful or questionable modifications, such as those caused by malware. If malware modifies the registry, it will alter the behavior of the system. By monitoring these edits, we may identify and rectify possible security vulnerabilities.

• **TCP dumps** are comprehensive logs of network traffic that provide specific information such as IP addresses, ports, and the volume of data being transmitted. An Intrusion Detection System (IDS) is utilized to oversee network activities and detect any potentially suspicious conduct. TCP dumps allow for the collection of data that accurately depicts the network's activity, enabling the detection and response to potential threats. They detect potentially illicit behaviors by examining trends in the data, such as abnormal surges in network activity, unforeseen associations with unknown IP addresses, or data transfers that differ from typical behavior. This analysis enables security professionals to promptly identify dangers and implement suitable measures to safeguard the network.

The term "number of sessions" denotes the frequency at which a user or application engages with a system. Session count tracking in an Intrusion Detection System (IDS) aid in identifying abnormal behavior, such as abrupt surges that could potentially signify a cyber-attack. Through the examination of session patterns, we can identify compromised accounts or potential security risks. If an account displays atypical session activity, such as concurrent logins from several places, it may suggest that the account has been compromised.

In IDS, converting the app into the binary format is very essential for security. This method ensures that the app data does not change, and this makes it easier to identify unauthorized changes. The use of binary formats allows IDS to easily compare the present state of any app with its previously known version and will hence identify anomalies that could indicate a threat. In case an app has a different binary representation than expected, malicious changes may be indicated. Binary formats enhance the effectiveness of the IDS so that they can maintain their level of security and expose the unauthorized changes.

The K-Means clustering algorithm was employed for this Intrusion Detection System (IDS) due to its robustness, versatility, scalability for big data, speed, and efficiency. The steps involved in building a model include:

- We partition the data into k similarity clusters based on specific criteria.
- Similarity Check: Use Euclidean, Manhattan, Hamming, and Cosine distances to measure similarity.
- Fixing Centroids: Select K random data points as the initial cluster centers.
- Assign each data point to nearest kth centroid.
- Calculate the cluster's centroids from all points.

Elbow method:
Core Points: The within-cluster sum of squares (WCSS) is computed as measure of cluster compactness.
$WCSS=\Sigma(\Sigma distance(Di,Ck))$
C represents the cluster centroids, while D represents the datapoints within each cluster.
Preprocessing techniques used in K-Means:

(i) Dealing with Missing Values

- Infinite values in the dataset were replaced with NaN.

All missing values in numerical columns were filled using SimpleImputer's mean technique

(ii) Re-Imputation of Missing Values

- After removing outliers, missing data were re-imputed using the mean method.

(iii) Normalization

- Standardized the dataset with StandardScaler to ensure equal contribution of all features to distance computations in K-Means clustering.

(iv) Dimensionality Reduction (for visualization)

- PCA was used to decrease dimensions to two, enabling display of clusters in a two-dimensional space.

Identifying the optimal number of clusters involves plotting the WCSS curve against the cluster numbers, with the bend in the plot indicating the estimated cluster count. The Silhouette Score curve is then plotted against the number of clusters, and the highest graph value determines the ideal K value.

IV. RESULTS AND DISCUSSION

Why K-Means clustering is Preferred for IDS:

1. To discover anomalies, we utilize an unsupervised learning technique without factor reduction. K-Means is the most effective clustering algorithm.
2. Robust: K-means can handle outliers and noise effectively.
3. Versatile: K-Means can handle many sorts of data.
4. Scalable for Big Data: K-Means excels at handling enormous datasets, making it ideal for networks with high traffic.
5. Fast and efficient: K-Means is easy to implement. K-Means is a cost-effective and efficient algorithm for processing huge datasets.
6. Interpretable: We can assess the data's organization.

Identifying the optimal number of clusters. Plot the WCSS curve based on cluster number. The location of the bend in a plot indicates the estimated number of clusters.

We plot the Silhouette Score curve against the number of clusters and consider the highest graph value as the K value.

Limitations of DBSCAN:

DBSCAN, or Density-Based Spatial Clustering of Applications with Noise, effectively addresses K-Means limitations by reducing sensitivity to initial centroid placement and identifying non-spherical cluster shapes common in network traffic datasets. Combining these algorithms creates a powerful and adaptable clustering method suitable for both simple and complex data.

To optimize DBSCAN's efficiency, we analyzed a random sample of 10,000 examples from the dataset. With parameters set at eps = 0.5 and min_samples = 10, our findings highlighted the algorithm's strong performance in detecting outliers and managing clusters with varying noise and density, as shown in the confusion matrix..

```
Confusion Matrix:
[[   0    0    0]
 [4370   11    0]
 [5619    0    0]]
```

Row 0 indicates instances predicted as cluster 0 (noise/outliers).

Row 1 corresponds to instances predicted as cluster 1.

Row 2 corresponds to instances predicted as cluster 2.

Columns:

Column 0 depicts true labels for cluster 0 (noise/outliers).

Column 1 depicts true labels for cluster 1.

Column 2 depicts true labels for cluster 2.

The contents of this table represent the number of instances that each row is assigned into a certain cluster Notice:

The value 4370 under Row 1, Column 0 indicates that 4370 those whose true label is cluster 0 fall into cluster 1.

Entry 11 says that only 11 samples of Row 1, Column 1 get correctly classified to cluster 1.

Entry 5619 says that 5619 samples from cluster 0 are wrongly classified to cluster 2.

There are several important facts depicted in this confusion matrix analysis:

This DBSCAN technique suffers from crucial issues while trying to separate the clusters with high rates of misclassifications.

Cluster 1 had the highest accuracy, only classifying 11 right out of 4381, a low 2.5% accuracy.

The number of misclassifications between cluster 0 and cluster 2 suggests the two clusters are not well-separated. From these results, it appears that DBSCAN is perhaps not the best clustering algorithm for this type of data set and the hyperparameters of eps and min_samples could be relaxed to further optimize the clustering.

Limitations of Isolation Forest:

These methods frequently struggle to identify outliers accurately in extensive datasets characterized by high variation and, hence, intricate distributions. High-dimensional data cannot be effectively scaled in application, and the presumption of equal feature weighting may be invalid, leading to inferior performance.

Limitations of GMM Algorithm:

```
Accuracy: 0.3694104407848975
Precision: 0.5373068767715685
Recall: 0.3694104407848975
F1 Score: 0.43773532336259224
Classification Report:
              precision    recall  f1-score   support

           0       0.19      0.14      0.16     97686
           1       0.80      0.55      0.65    128025
           2       0.00      0.00      0.00         0

    accuracy                           0.37    225711
   macro avg       0.33      0.23      0.27    225711
weighted avg       0.54      0.37      0.44    225711

Confusion Matrix:
[[13529 17140 67017]
 [58001 69851   173]
 [    0     0     0]]
```

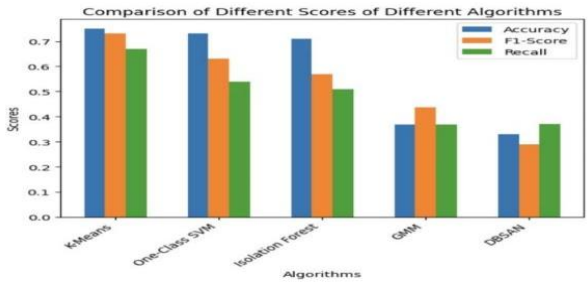
The results of our code indicate that the classification metrics reveal significant imbalance in the performance of GMM clusters.

Class '2' lacks predictions, resulting in a zero row in the confusion matrix, which adversely affects the macro average, leading to diminished precision and recall. The model appears to exhibit bias towards class '1', since its predictive performance for the other class is suboptimal. This may be attributed to the over-representation of one class and the likely under-representation of the other, potentially stemming from issues with clustering.

Limitations of One-Class SVM:

Another restriction SVMs have is a time complexity of $O(n^2)$ or worse, which makes it hard to scale for larger datasets; training times grow very quickly as the size of your dataset does. In principle, linear SVMs are faster, but they may not be strong enough to detect complex patterns of dependency and take advantage of the good properties of kernel functions that describe nonlinear relationships.

Algorithm	Accuracy	f1-Score	Recall
K-Means	0.75	0.73	0.67
One-Class SVM	0.73	0.63	0.54
Isolation Forest	0.71	0.57	0.51
GMM	0.369	0.4377	0.369
DBSCAN	0.33	0.29	0.37



V. CONCLUSION

K-means is a popular method for conducting cluster analysis that seeks to partition 'n' data objects into 'k' clusters, with each data object assigned to the cluster corresponding to the nearest mean. It is a distance-based clustering method that does not need the calculation of distances between all record combinations. This study presents a survey of data-mining techniques utilized in the construction of intrusion detection systems. [4]

Assessment of Intrusion Detection System Efficacy:

The IDS attained an overall accuracy of 0.75. The confusion matrix indicates that the KMeans clustering technique markedly enhances the distinguishing capability of the Intrusion Detection System (IDS) between benign traffic and DDoS attacks. The decrease in both false positives and false negatives, in comparison to earlier models, demonstrates that K-Means clustering independently contributes significantly to improving the accuracy of an Intrusion Detection System (IDS).

Accuracy:

Accuracy is the ratio of correct classifications to the total number of classifications, regardless of whether they are positive or negative. It can be mathematically expressed as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Recall, or true positive rate:

True positive rate (TPR), which represents the fraction of all actual positives correctly identified as positives, is also referred to as recall.

$$\text{Recall (or TPR)} = \frac{TP}{TP+FN} \quad (2)$$

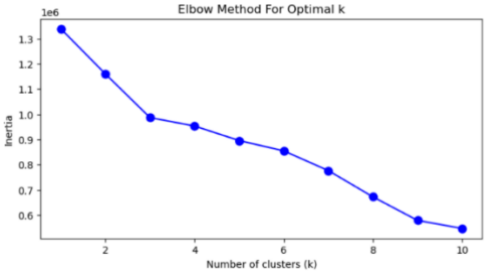
Precision:

Precision refers to the ratio of the model's positive predictions that are positive. It can be mathematically expressed as:

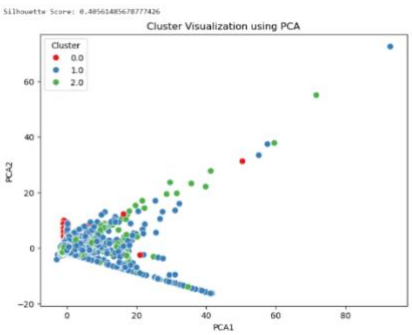
$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

```
True Labels
BENIGN          19947    77739
DDoS            58185    69840
              precision    recall  f1-score   support

      BENIGN       0.64      0.68      0.62     97686
        DDoS       0.67      0.69      0.73    128025

    accuracy                           0.75    225711
```

All the test is done for the result of Label column which has two outputs one is BENIGN and other is DDOS attack identified



Through the Elbow Method for optimal K, we can say that optimal K Value would be 3.



The silhouette score of 0.40 indicates a satisfactory value, suggesting that the clusters are well-formed. However, there is some overlap among the clusters, which implies that minor changes could lead to suspicious activities, making it challenging to distinctly separate the clusters for the given dataset. The blue circles represent the safe zone labeled as BENIGN, while the limited presence of red and green points indicates ongoing suspicious activities that require attention. The best number of clusters is determined with the help of the Elbow Method, considering the k corresponding to the moment when the WCSS curve is flattening (k=3). For instance, when the value of k was set to 3, the resulting Silhouette Score was 0.49, pointing to moderate cohesion and separation between clusters. The score was not especially high, but it did show that the clusters had been reasonably well-formed. Combining information from both methods, it was determined that k=3 (the optimal number of clusters) was a good fit between cohesion and separation.

VI. FUTURE WORKS

Improvement can be achieved in IDS through the refinement of inference mechanisms in the rule-based frameworks as well as the inclusion of machine learning that would allow for automatic anomaly detection. Therefore, scalability needs to be enhanced so that large datasets can be handled, with real-time detection implemented concurrently with automatically triggered countermeasures to address resource and mobility constraints that are known to hinder the security of mobile systems. Most importantly, safe protection of users' privacy through encryption and privacy-preserving techniques remain intact in safeguarding sensitive data.

Advances in the IDS of the future could benefit much from integrating AI and ML for advanced detection of sophisticated and previously unknown threats and for the reduction of false positives. Scalability is one of the key requirements for large-scale real-time network monitoring, and hence the architectures require to be optimized and scalable in nature. Cloud-native IDS solutions and lightweight systems especially for Internet of Things devices are also sought for addressing emerging trends in technological dimensions. Sophisticated techniques for clustering will enable better detection mechanisms against complex attacks using techniques such as refined K-Means Clustering. Automation and autonomous responses will embrace self-healing networks, which include integrating various security tools to develop a more holistic defense posture. Finally, research into quantum-resistant IDS will play an important role in countering any threat from future quantum-based attacks; thus, developing an infrastructure that will help keep that digital world secure.

REFERENCES

[1] Ashawa, M., & Morris, S. (2019). Analysis of Android Malware Detection Techniques: A Systematic Review. *International Journal of Cyber-Security and Digital Forensics*, *8*(3), 177–187.

[2] Song, J., Choi, S., Kim, J., Park, K., Park, C., Kim, J., & Kim, I. (2024). A study of the relationship of malware detection mechanisms using Artificial Intelligence. *ICT Express*.

[3] *What is an Intrusion Detection System (IDS)? | IBM*. (n.d.).

[4] Mohammed, M. S., & Talib, H. A. (2024). Using Machine Learning Algorithms in Intrusion Detection Systems: A Review. *Tikrit Journal of Pure Science*, *29*(3), 63–74.

[5] Mohammed, M. S., & Talib, H. A. (2024). Using Machine Learning Algorithms in Intrusion Detection Systems: A Review. *Tikrit Journal of Pure Science*, *29*(3), 63–74.

[6] Mohammed, M. S., & Talib, H. A. (2024). Using Machine Learning Algorithms in Intrusion Detection Systems: A Review. *Tikrit Journal of Pure Science*, *29*(3), 63–74.

[7] Musa, U., Chhabra, M., Ali, A., & Kaur, M. (2022). Intrusion Detection System using Machine Learning Techniques: A Review. *ResearchGate*.

[8] Ahmed M, Mahmood AN, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl. 2016;60:19–31.

[9] Modi, C. (2014). A survey of intrusion detection techniques in Cloud. *Svnit*.

[10] Riaz, S. (2021). Intercept the Cloud Network From Brute Force and DDoS Attacks via Intrusion Detection and Prevention System. *www.academia.edu*.

[11] S. Kumar, S. Gupta and S. Arora, "Research Trends in Network-Based Intrusion Detection Systems: A Review," in *IEEE Access*, vol. 9, pp. 157761-157779, 2021, doi: 10.1109/ACCESS.2021.3129775.

[12] Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, *2*(1).

[13] Liao, H., Lin, C. R., Lin, Y., & Tung, K. (2013). Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), 16–24.

[14] Hoque, M. S., Mukit, M. A., & Bikas, M. a. N. (2012). An Implementation of Intrusion Detection System Using Genetic Algorithm. *International Journal of Network Security & Its Applications*, *4*(2), 109–120.

[15] M. A. Qadeer, A. Iqbal, M. Zahid and M. R. Siddiqui, "Network Traffic Analysis and Intrusion Detection Using Packet Sniffer," *2010 Second International Conference on Communication Software and Networks*, Singapore, 2010, pp. 313-317, doi: 10.1109/ICCSN.2010.104.

[16] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," in *IEEE Access*, vol. 7, pp. 41525-41550,2019,doi:10.1109/ACCESS.2019.2895334

[17] Biermann, E., Cloete, E., & Venter, L. (2001). A comparison of Intrusion Detection systems. *Computers & Security*, *20*(8), 676–683.

[18] Javaid, A., Niyaz, Q., Sun, W., & Alam, M. (2016). A Deep Learning Approach for Network Intrusion Detection System. *Jk*.