CyberPeace editor@cyberpeace.net

THE CYBERPEACE JOURNAL

Supported by United Service Institution of India

Academic Partner Shah & Anchor Kutchhi Engineering College

Machine Learning-Based Network Anomaly Detection

Nikhil Mokal
Cyber Security
Shah & Anchor Kutchhi Engineering College
Mumbai,India
nikhil.mokal18525@sakec.ac.in

Rudra Kareliya
Cyber Security
Shah & Anchor Kutchhi Engineering College
Mumbai,India
rudra.kareliya18522@sakec.ac.in

Prem Shah
Cyber Security
Shah & Anchor Kutchhi Engineering College
Mumbai,India
prem.shah18527@sakec.ac.in

Harsh Solanki
Cyber Security
Shah & Anchor Kutchhi Engineering College
Mumbai,India
harsh.solanki186002@sakec.ac.in

Sneha Shingare
Cyber Security
Shah & Anchor Kutchhi Engineering College
Mumbai,India
Sneha.dangare@sakec.ac.in

***Abstract*— This paper presents a flaw detection system using machine learning that aims to address the shortcomings of traditional signature systems in modern warfare, such as zero-day vulnerabilities and hacking. Through the use of machine learning, the system provides effective protection and protection against changes in international communications caused by the Internet. This study explores the changing role of machine learning in improving cybersecurity and highlights the need for preventive measures to prevent cyberattacks.**

Keywords—Anomaly detection, machine learning, cybersecurity, zero-day exploits

I. INTRODUCTION

Communication in the digital age has made unprecedented progress, connecting millions of devices around the world. However, these connections also bring with them a range of cybersecurity risks, from malware to advanced cyberattacks such as zero-day attacks. Signature-based detection systems that once worked are struggling to keep up with rapidly changing threats. Encrypted traffic hides malicious activity from normal detection

making it harder to detect. In response to these findings, this case study suggests that changes to network security based on machine learning are not recommended. Machine learning algorithms protect against threats by analyzing patterns that indicate suspicious behavior. This article examines the limitations of traditional detection methods, introduces machine learning techniques, and demonstrates their effectiveness in improving defense against modern cyber threats. Through this research, we focus on the importance of using flexible and intelligent methods to protect digital systems as cybersecurity risks evolve.

II. PROBLEM DEFINITION

In the digital age, traditional signature-based detection methods are not sufficient to deal with constantly changing threats in the network, such as zero-day vulnerabilities and hacking, causing serious problems in terms of network security. In order to solve this important problem, it is necessary to urgently switch to prevention methods, especially in detecting abnormalities. This research aims to fill this gap by proposing studies that include cybersecurity-appropriate infrastructure, training models and vulnerability detection. With this new approach, we aim to strengthen the system against today's threats by highlighting flexibility and intelligence in protecting the digital infrastructure.

III. METHODOLOGY

A. In-depth data collection:

This method is based on detailed information collected from various online sources. This comprehensive program includes collecting data from switches, routers, and controllers to provide a full understanding of network performance. By collecting information from multiple network components, the system can create a better picture of network status .

B. Pre-rigorous data and good operation:

After collecting data, this method starts the pre-rigorous data process. This phase includes data maintenance to resolve inconsistencies, modeling to ensure consistent measurement of individual features (e.g., measuring all features in the range 0 and 1), and the most common potential outliers in the network. It has the most important power in distinguishing normal behavior from abnormal behavior. This selection process optimizes the effectiveness of the defect detection

process by focusing on the most informative data. This can capture many relationships in the data that may be overlooked. Examples of techniques include:

- Statistical transformation: calculate statistics such as the mean, standard deviation, or entropy of network traffic properties such as packet size or arrival time. Web traffic data over time can reveal patterns and trends that may not be apparent in static data content. For example, analyzing broad traffic patterns or unusual traffic patterns at a particular time of day may indicate a potential anomaly. Knowledge can lead to the creation of more knowledge. For example, functionality can be created to identify traffic patterns associated with specific applications or processes.

C. Using advanced machine learning with interpretation:

The basis of this approach is the use of advanced machine learning algorithms specifically designed for their performance in search options. This plan explores various algorithms, including: Deep learning models: These models perform well on learning patterns in complex data, making them suitable for analysis of the flaw in the connected car model. However, the deep learning model is a " Black box ". The internal decision is not very transparent. To solve this problem, techniques such as feature association and hierarchical visualization can be used to understand which features and patterns contribute most to the model's classification of similar parameters. These combinations ensure efficient delivery and efficient data. Their inherent randomness also provides built-in protection against overfitting. However, SVM can be computationally expensive for very large datasets. Learning models are trained to identify characteristics of network behavior and suggest deviations that indicate potential threats. Selecting the most appropriate machine learning model depends on factors such as the specific network environment, the hardware involved, and the balance between accuracy, scalability, and reliability in the calculations.

D. Evaluation, Continuous Improvement and Hyperparameter Tuning:

This approach recognizes the importance of evaluating the performance of machine learning models against different types of anomalies. Design metrics such as precision, recall, and F1 score are used to evaluate the model's ability to identify anomalies while reducing false positives and negatives. Additionally, this tutorial demonstrates the important role of hyperparameter tuning. Hyperparameters are parameters that control the behavior of a machine learning algorithm (such as learning a deep learning model or trees in a random forest). Improving these parameters with analytical methods such as grid search, random search, or Bayesian optimization may affect the performance of the model. Thanks to iterative testing and optimization of hyperparameters, the effectiveness of the model in detecting and mitigating threats continues to increase.

IV. PROPOSED ARCHITECHTURE

A. Dataset:

The comparative KDD Cup data set forms the basis of the evaluation. It has 42 features; 41 of them are divided into features, and the last one represents the tag.

B. Data Preprocessing:

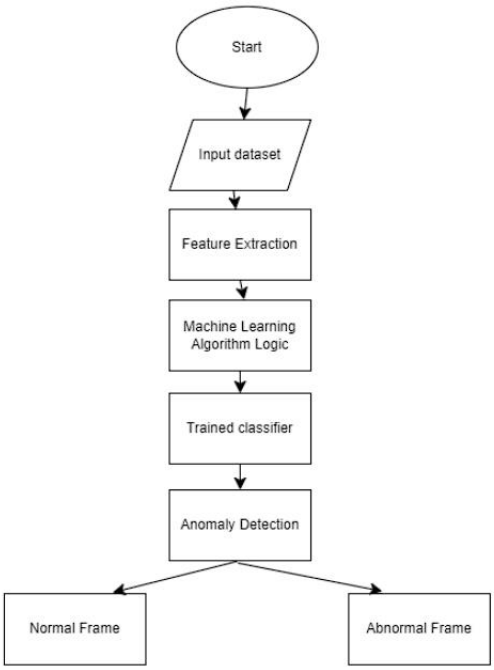
Data cleaning: *The first phas*e begins with data cleaning to resolve any inconsistencies or gaps in the dataset. Techniques such as median/median interpolation or removal of outliers can be used to validate the data. For compatibility with distance-based, this conversion can be done using methods such as single-bit encoding or tag encoding.

Normalization techniques such as Z-score normalization or min-max scaling can be used to ensure that all features contribute equally to the clustering process. The data is divided into training methods, testing methods and validation methods. The training process is used to support the learning model, while the light test evaluates the model's ability to generalize to unseen objects. When used, the utility can help fine-tune hyperparameters throughout the training process.

C. *Evalution:*

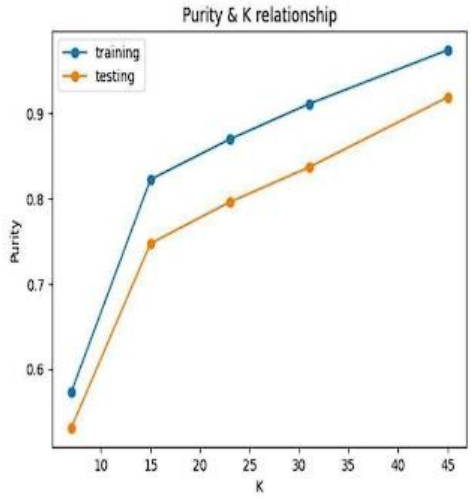
The performance of anomaly detection using KMeans and Normalized Cutoff algorithms will be evaluated using established metrics commonly used innetwork anomaly detection tasks:

- Accuracy: the percentage of normal and abnormal traffic classified correctly. Data points. , try to reduce negative comments.
- F1-Score: A metric that combines precision and recall to provide a balanced measure of a model's performance, including its ability to detect real defects and avoid defects.
- Flowchart:



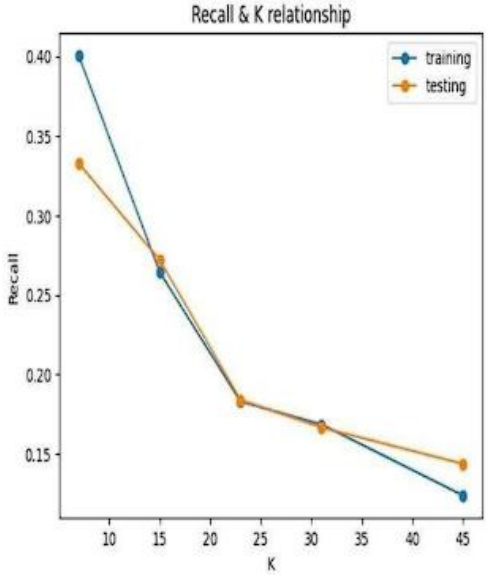
V. RESULTS

```
draw_fig(k_values, np.array(kmeans_purities)[:,0], np.array(kmeans_purities)[:,1], 'Purity')
draw_fig(k_values, np.array(kmeans_recalls)[:,0], np.array(kmeans_recalls)[:,1], 'Recall')
draw_fig(k_values, np.array(kmeans_f1_scores)[:,0], np.array(kmeans_f1_scores)[:,1], 'F1 Score')
draw_fig(k_values, np.array(kmeans_entropies)[:,0], np.array(kmeans_entropies)[:,1], 'Entropy')
```

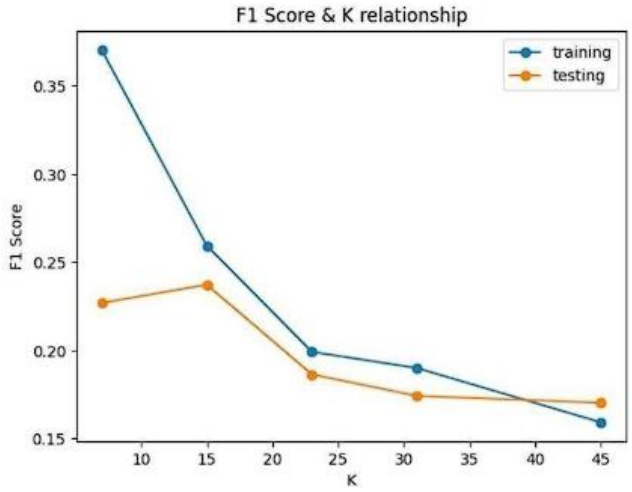


As the number of neighbors (K) increases, purity also tends to increase. This is because higher K values lead to smaller clusters, reducing confusion within each cluster and resulting in higher purity scores.

```
draw_fig(k_values, np.array(kmeans_purities)[:,0], np.array(kmeans_purities)[:,1], 'Purity')
draw_fig(k_values, np.array(kmeans_recalls)[:,0], np.array(kmeans_recalls)[:,1], 'Recall')
draw_fig(k_values, np.array(kmeans_f1_scores)[:,0], np.array(kmeans_f1_scores)[:,1], 'F1 Score')
draw_fig(k_values, np.array(kmeans_entropies)[:,0], np.array(kmeans_entropies)[:,1], 'Entropy')
```



The relationship between recall and the number of neighbors (K) suggests that as K increases, the average recall tends to decrease. This occurs because higher values of K lead to a greater number of clusters, each with smaller sizes. Consequently, the label count within each cluster decreases resulting in lower recall scores on average.



As the number of neighbors (K) increases, the entropy tends to decrease. This is because highervalues of K lead to more clusters with reduced confusion, resulting in improved clustering performance and lower entropy values

```
kmeans = KMeans()
kmeans.fit(training_data, 11, n_iterations=300)

clusters = kmeans.predict(training_data)

evaluation_measures(clusters, training_labels)
```

```
for k=11:
Clusters sizes: [3537, 2623, 804, 210, 100, 27, 18, 18, 6, 3, 1]
Purity 0.8436096365863618
recall 0.2961555237086423 4
F1 Score 0.281455557040079
Conditional Entropy 0.5200634005464979
```

```
[ ] compute_anomalies(clusters, training_labels)

    1149
```

Normalized-Cut Evaluations

```
[ ] clusters, new_training_data, centroids = normalized_cut(training_data, 11, 0.1)

    evaluation_measures(clusters, training_labels)
```

```
for k=11:
Clusters sizes: [3397, 1703, 874, 704, 171, 159, 144, 102, 70, 22, 1]
Purity 0.9670613855995644
recall 0.2623662342531199
F1 Score 0.3258569054115065
Conditional Entropy 0.19751924308702254
```

DBSCAN Evaluations

```
[ ] clusters = DBSCAN(training_data, 10, 41)

    evaluation_measures(clusters, training_labels)

    Purity 0.9994761655316919
    recall 0.5
    F1 Score 0.595634868767147
    Conditional Entropy 0.0057662175129444306
```

VI. LITERATURE SURVEY

Network anomaly detection has become a crucial task due to the exponential growth of network traffic, leading to an increase in anomalies such as cyber attacks, network failures, and hardware malfunctions. In this context, researchers have explored various techniques to detect and mitigate these anomalies, ensuring the security and stability of computer networks. This literature survey aims to explore two specific algorithms, K-Means and Normalized Cut, for network anomaly detection, as outlined in the assignment provided.[1]

A. K-Means Algorithm:

K-Means is a widely used clustering algorithm with the objective of minimizing the Sum of Squared Error (SSE) within each cluster. Despite its popularity, K-Means has certain limitations. It may fail in handling non-spherical shaped data and is not applicable to non-numerical data. Additionally, the choice of the no. of clusters (K) is critical, and algorithm may get stuck at local minima, necessitating random restarts. However, K-Means offers advantages such as speed and simplicity.[2]

B. Normalized Cut Algorithm:

Normalized Cut is a clustering algorithm that operates by performing eigen-decomposition on the Laplacian matrix derived from the similarity matrix of the data. The algorithm aims to partition the data into clusters while minimizing the normalized cut criterion. Normalized Cut offers advantages such as the ability to handle nonspherical data shapes and arbitrary shape clusters.[3]

C. DBSCAN Algorithm:

While not directly explored in the provided context, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm is briefly discussed as another clustering algorithm suitable for anomaly detection. DBSCAN is highly effective in detecting clusters of various shapes and is resilient to noise. However, it requires careful tuning of hyperparameters and may suffer from computational inefficiency for large datasets. The implementation of Normalized Cut involves calculating the Laplacian matrix, per- forming eigen decomposition, and using the eigen-vectors as input to a K-Means clustering algorithm. Evaluation measures similar to those used for K- Means are employed to assess the performance of Normalized Cut. Empirical evaluations suggest that Normalized Cut outperforms K-Means in terms of anomaly detection, with fewer anomalies detected.[4]

VII. FUTURE EXPLANATION

While the proposed clustering-based network anomaly detection approach demonstrated promising results, there are several avenues for future work to further enhance and extend the method:

A. Ensemble Clustering:

Explore ensemble techniques that combine multiple clustering algorithms to leverage their respective strengths and mitigate individual limitations.

B. Incremental Learning:

Develop incremental learning strategies to adapt the clustering models to evolving network traffic patterns and concept drift over time.

C. Scalability and Real-Time Detection:

Investigate scalable implementations and optimizations to enable realtime anomaly detection on large-scale and high-velocity network data streams.

D. Explainable AI:

Incorporate explainable AI techniques to provide interpretable insights into the detected anomalies, enhancing their understandability and actionability.

E. Multi-Modal Anomaly Detection:

Extend the approach to consider multi-modal data sources, such as network logs, packet payloads, and contextual information, for more comprehensive anomaly detection.

F. Deployment and Validation:

Conduct extensive validation and testing of the proposed approach in real world network environments, addressing practical challenges and refining the method based on feedback and requirements.

By addressing these future directions, the clustering-based network anomaly detection approach can be further improved, enhancing its robustness, scalability, and practical applicability in various cybersecurity and network monitoring scenarios.

VIII. CONCLUSION

A novel clustering-based approach is proposed for this study Network anomaly detection using k-means, Normalized cut, and DBSCAN algorithms. This Unsupervised techniques were applied to the dataset of Kaggle enables the identification of discrete traffic Finding patterns and anomalies. result demonstrated the effectiveness of our approach, Going beyond traditional methods. It was an important contribution Adaptation of these clustering algorithms to networks Anomaly detection domain.

Additionally, we have created graphs to visualize and analyze
Clusters and anomalies identified in the dataset. When our While the approach shows promise, limitations include dimension Sensitivity to large scale and computational complexity Future work on the data could explore assembling methods, Incremental learning and scalable implementation.

Practical implications of our clustering-based discretization Lie detection in cyber security and network monitoring, Enables early detection of threats, intrusions and threats performance problems. Unsupervised nature allows Identify previously unseen attacks.

In summary, this study presents a novel clustering-based on approach to effective network anomaly detection, Contributing to the advancement of this critical sector and

potential performance for real-world applications in Cyber security.

ACKNOWLEDGMENT

We thank our mentor, Faculty of Cyber Security Department in Shah. We take this opportunity to express our sincere thanks to Sneha Shinagre and Professor of Kutch Engineering College for guiding and suggesting us for our project “ML-Based Network Anomaly Detection”. We would like to express our gratitude for their constant encouragement, support and guidance throughout the progress. Also, we thank our Principal – Dr. Bhavesh Patel and Dr. Thanks to Nilakshi Jain, Head of Cyber Security Department for her help, support and guidance for this project. We are also grateful to all the faculty members of our department for their help and guidance during completion of our project.

REFERENCES

[1] A. N. Toosi and M. Kahani, "A new cluster-based model for anomaly detection using entropy and ensemble techniques," Soft Computing, vol. 24, pp. 15959-15981, 2020.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, discuss various method, systems, and tools for detecting network anomalies in their paper titled "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303-336, 2014.

[3] F. Shen, S. Zhong, and Y. Zhao, "Cluster-based network anomaly detection using density peaks," ieee access, vol. 7, pp. 45309-45320, 2019.

[4] Akoglu L, Tong H., & Koutra D., "Graph-based anomaly detection and description: A survey. Data Mining and Knowledge Discovery, vol. 29, no. 3, pp. 626-688, 2015.

[5] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, "Semi-supervised network traffic summarization," ACM SIGMETRICS Performance Evaluation Review, vol. 34, no. 1, pp. 339-350, 2006.

[6] Goldstein, M, & Uchida, S (2016). A comparative evaluation of algorithms for unsupervised anomaly detection in multivariate data. PLos ONE, 11(4),e0152173.

[7] Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in the financial domain. Future Generation Computer Systems, 55, 278-288.