CLUSTERS IN CHAOS: A DEEP UNSUPERVISED LEARNING PARADIGM FOR NETWORK ANOMALY DETECTION

Seethalakshmi Perumal*, P Kola Sujatha*, Krishnaa S† and Jawahar A S†
Department of Information Technology, Madras Institute of Technology Campus, Anna University Chennai, Tamil Nadu, India
* seethalaxmiperumal@gmail.com; pkolasujatha@annauniv.edu † krishnaa2902@gmail.com; asjawahar05052003@gmail.com

***Abstract*—In response to the escalating sophistication of cyber threats, traditional security measures are proving insufficient, necessitating advanced solutions. The complexity of cyberattacks renders standard protocols inadequate, leading to an increased frequency of disruptions, data breaches, and financial losses. To address this challenge, the development of intelligent systems capable of recognizing unusual network activity has become imperative. These cutting-edge technologies, employing deep learning techniques, analyze vast volumes of real time network data to identify patterns and anomalies, enabling early cyber threat identification. This paradigm shift towards proactive and adaptable cybersecurity approaches is crucial in today's dynamic and complex cyber threat landscape.**

The research focuses on the implementation of intelligent systems, emphasizing the analysis of the NSL KDD dataset. This involves a thorough exploration of diverse network traffic, attributes, and attack scenarios to gain comprehensive insights. The development of a deep unsupervised learning model further enhances the system's capability to detect subtle anomalies within the dataset. A novel approach utilizing Autoencoder based clustering is introduced, involving the grouping of similar normal data points through clustering, followed by training individual autoencoders for each cluster. This innovative technique captures nuanced patterns of normal behavior within each cluster, significantly enhancing the model's ability to discern anomalies. These advancements over standard methods signify a substantial leap forward in anomaly detection, addressing limitations, improving accuracy, and showcasing versatility in handling diverse network traffic scenarios.

***Index Terms*—Autoencoder, clustering, K-Means, NSL-KDD, Network Anomaly**

I. Introduction

The rapidly evolving landscape of digital connectivity has made network anomalies a significant threat to the stability and security of information systems. Network anomalies, which can result from cyberattacks, system malfunctions, or configuration errors, have the potential to cause severe disruptions, including prolonged downtime, data breaches, financial losses, and compromised sensitive information. Such incidents underscore the urgent need for robust mechanisms to detect and mitigate these anomalies early. The increasing complexity and frequency of these anomalies demand sophisticated detection systems capable of identifying threats in real-time to prevent catastrophic consequences.

Advanced detection systems leveraging autoencoders and clustering techniques have become indispensable in addressing these challenges. Autoencoders, a class of unsupervised deep learning models, are particularly effective in network anomaly detection due to their ability to learn compressed representations of normal network behavior and identify deviations indicative of potential threats. For instance, the Paper [1] by Lunardi et al., utilizes a convolutional autoencoder to automatically learn normal traffic patterns from network flow packets. This system enhances anomaly detection capabilities by employing adversarial training to constrain the autoencoder's ability to reconstruct out-of-normal distribution flows, demonstrating superior effectiveness in detecting malware infections and network attacks .

In addition to autoencoders, clustering techniques play a crucial role in refining the anomaly detection process. Clustering allows for the categorization of network data into distinct groups based on inherent similarities, facilitating the identification of outliers that may signify anomalies. Aytekin et al. in Paper [2] have shown that applying an L2 normalization constraint during autoencoder training significantly improves clustering accuracy and anomaly detection performance. Their study highlights that L2 normalized deep autoencoder representations lead to more separable clusters in Euclidean space, enabling simple k-means clustering to achieve high accuracy without additional clustering losses . This approach further underscores the utility of unsupervised deep learning models in enhancing both clustering and anomaly detection tasks in dynamic network environments.

Integrating these techniques ensures a proactive and resilient approach to safeguarding digital infrastructure against an ever-evolving spectrum of cyber threats. The implementation of autoencoders and clustering in unsupervised anomaly detection systems represents a paradigm shift towards more adaptable and intelligent cybersecurity solutions. This shift is critical in today's complex cyber threat landscape, where traditional security measures often fall short.

The following chapters explore more into the proposed work. Chapter 2 gives the literature survey with it's limitations. Chapter 3 explores the proposed work starting with the Overall Architecture. Chapter 4 explains the different sub modules that are present in the system. The results and discussion of the proposed work is done in Chapter 5. Finally, the conclusion and Future work part of the work is present in Chapter 6.

II. Literature Survey

The paper [1] by Lunardi et al. introduced ARCADE (Adversarially Regularized Convolutional Autoencoder for unsupervised network anomaly DEtection), a practical unsupervised anomaly-based deep learning detection system designed to enhance network security in the face of increasing heterogenous IP-connected devices and traffic volumes. ARCADE utilizes a convolutional Autoencoder (AE) to automatically learn normal traffic patterns from a subset of raw bytes of initial network flow packets, enabling efficient detection of potential anomalies and intrusions before they escalate. The system is trained exclusively on normal traffic and employs an adversarial training strategy to constrain the AE's ability to reconstruct out-of-normal distribution flows, thereby enhancing its anomaly detection capabilities. ARCADE demonstrates superior effectiveness compared to existing deep learning approaches, detecting malware infections and network attacks even with minimal input data. Notably, ARCADE boasts 20 times fewer parameters than baselines, resulting in significantly faster detection speeds and reaction times. Experimental results reveal that ARCADE achieves nearly 100% F1-score for most malicious traffic types, except for HTTP DoS and DDoS attacks where it reaches 68.% and 66.61% F1-scores, respectively. Furthermore, when considering five packets, ARCADE achieves even higher F1-scores of 91.95% and 93.19% for HTTP DoS and DDoS attacks, respectively, demonstrating its efficacy in improving accuracy, model simplicity, and detection speed.

The paper by Aytekin et al. investigates the role of L2 normalization constraint in enhancing the separability and compactness of representations learned by deep auto-encoders for clustering analysis and unsupervised anomaly detection [2]. It demonstrates that applying an L2 normalization constraint during auto-encoder training improves the clustering accuracy significantly when employing k-means clustering on the learned representations. Additionally, the authors proposed a novel clustering-based unsupervised anomaly detection method utilizing L2 normalized deep auto-encoder representations, showcasing its effectiveness in anomaly detection accuracy compared to existing deep methods like reconstruction error-based approaches.

The study highlights that the L2 normalization constraint applied to deep auto-encoder representations leads to more separable clusters in the Euclidean space, enabling simple k-means clustering to achieve high accuracy without the need for additional clustering losses. The observed performance improvement is attributed to the specific normalization method chosen, rather than any conditioning applied to the representations. The proposed unsupervised anomaly detection method builds upon these normalized representations, further emphasizing the utility of L2 normalization in enhancing both clustering and anomaly detection tasks in deep learning contexts.

Maudox et al. proposed an approach for detecting network anomalies by aggregating pre-processed network flows into sectors, dividing data into equal time periods, and employing unsupervised clustering to extract activity deviations [3]. By identifying deviations in network activity within specific sectors and time periods, the method successfully detects anomalies corresponding to real-world events like crowded gatherings. Leveraging an unsupervised machine learning algorithm, the approach characterizes network behaviors and extracts outliers to pinpoint anomalies, demonstrating its effectiveness in event detection and anomaly identification.

The study by Park et al. introduced a novel AI-based Network Intrusion Detection System (NIDS) aimed at addressing the data imbalance problem prevalent in existing systems [4]. By leveraging state-of-the-art generative models, including reconstruction error and Wasserstein distance-based generative adversarial networks, alongside autoencoder-driven deep learning models, the NIDS proposed by the authors generates synthetic data to supplement minor attack traffic, enhancing classification performance. Comprehensive evaluations across various datasets, including benchmark, IoT, and real-world data, demonstrate significant performance improvements, with accuracies reaching up to 93.2% and 87% on NSL-KDD and UNSW-NB15 datasets, respectively. Additionally, the proposed NIDS showcases efficient detection of network threats in distributed environments and real-world enterprise systems, laying groundwork for future research on federated learning systems, ensemble AI systems, and enhanced NIDS to combat adversarial attacks.

The paper by Bennadi et al. addressed the escalating vulnerability of modern systems to cyber-attacks due to advancements in networking technologies [5]. With a surge in security incidents and network activities, the focus has shifted towards developing Intrusion Detection Systems (IDSs) predominantly based on traditional machine learning and deep learning models. Recognizing the remarkable performance achieved in diverse fields through Deep Reinforcement Learning (DRL), which combines deep learning with reinforcement learning, the authors introduced a novel DRL-based IDS for network traffic analysis, leveraging Markov decision processes (MDP) to enhance decision-making. Furthermore, the study delves into the intricate dynamics between the IDS and attackers by employing Stochastic Game Theory, modeling the interaction through a non-zero-sum stochastic game to reach a Nash Equilibrium. This equilibrium ensures optimal decision policies where both parties maximize their gains, ultimately improving network security. Evaluations conducted on the NSL-KDD dataset demonstrate superior performance of the proposed DRL-IDS compared to existing models, manifesting in enhanced detection rates, accuracy, and reduced false alarms.

The paper by Ursul et al. offered a comprehensive review of anomaly detection algorithms with a focus on clustering applications across various domains such as user analysis, network intrusion detection, fraud detection, and system monitoring [6]. It extensively analyzes different clustering techniques including distance-based, hierarchical, and density-based methods, along with ensemble techniques and outlier detection methods to enhance anomaly detection accuracy. The authors applied these algorithms to a customer dataset to detect anomalies and compared their performance based on scalability, precision, recall, and F1-score metrics. Furthermore, the authors addressed challenges in cluster-based anomaly detection such as selecting the appropriate number of clusters, handling high-dimensional data, and dealing with imbalanced datasets. Finally, it provides insights into overcoming these challenges and suggests future research directions. Additionally, the study presented a cluster-based unsupervised anomaly detection method specifically tailored for identifying anomalous running patterns in activity datasets, showcasing its efficacy in detecting various types of anomalies and physical activities.

The paper by Ambusaidi et al. addressed the challenge of redundant and irrelevant features in network traffic classification, which hinder classification accuracy and increase computational complexity, especially with big data. To tackle this issue, the authors proposed a mutual information-based feature selection algorithm capable of handling both linearly and nonlinearly dependent data features [7]. They demonstrate the effectiveness of the algorithm in the context of network intrusion detection by integrating it into an Intrusion Detection System (IDS) called Least Square Support Vector Machine-based IDS (LSSVM-IDS).

Furthermore, the author introduced a supervised filter-based feature selection algorithm called Flexible Mutual Information Feature Selection (FMIFS), which is an improvement over existing methods like MIFS and MMIFS. FMIFS addresses redundancy among features by modifying Battiti's algorithm, eliminating the need for a redundancy parameter, thus simplifying the feature selection process. FMIFS is then integrated with LSSVM to construct an IDS, leveraging the robustness of LSSVM for classification tasks. Evaluation on three intrusion detection datasets KDD Cup 99, NSL-KDD, and Kyoto 2006+ illustrates that the proposed feature selection algorithm significantly enhances LSSVM-IDS's accuracy and reduces computational costs compared to state-of-the-art methods. This combination further enhances the efficiency and accuracy of the IDS, offering a promising solution for network intrusion detection.

The paper by Ravi et al. addressed the need for intrusion detection in IoT networks, which are vulnerable to security breaches that can compromise performance and data security [8]. Unlike existing solutions relying on supervised learning methods requiring large labeled datasets, the proposed approach introduces a novel SDRK machine learning algorithm. SDRK combines supervised deep neural networks (DNNs) with unsupervised clustering techniques and implements intrusion detection and mitigation algorithms in fog nodes between IoT devices and cloud infrastructure. The methodology is tested against data deluge (DD) attacks, with results showcasing improved accuracy of 99.78% when evaluated on the NSL-KDD dataset compared to state-of-the-art solutions. This innovative approach offers a promising solution for enhancing intrusion detection in IoT networks while overcoming challenges associated with labeled dataset availability.

The paper by Shone et al. introduced a novel approach to network intrusion detection systems (NIDSs) to address concerns about current methods' feasibility and sustainability in modern networks [9]. The author utilizes a nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning and a deep learning classification model constructed using stacked NDAEs. Implemented in GPU-enabled TensorFlow, the resulting classifier is evaluated on benchmark datasets (KDD Cup '99 and NSL-KDD), yielding promising results that surpass existing approaches. The innovative NDAE approach demonstrated potential for enhancing NIDS performance and addressing evolving network security challenges.

The study by Khan et al. addressed the escalating threat of malware in modern manufacturing, driven by the widespread adoption of vulnerable IoT technologies in Industry 4.0 [10]. To counter sophisticated attacks like zero-day exploits and Mirai botnets, the authors proposes a novel security mechanism combining Deep Autoencoder (DAE) and Support Vector Machine (SVM). Tested on the NSL-KDD dataset, known for its imbalanced, multi-class, and high-dimensional nature, the hybrid DAE-SVM scheme underwent grid search analysis for regularization and explored various neural network architectures to enhance classification metrics like F1-micro and balance accuracy. Evaluation across four attack classes demonstrated the superiority of DAE-SVM over PCA-SVM, particularly in detecting low-frequency attacks. Additionally, the study investigates optimal feature fusion strategies for minimizing computational overhead, with DAE-SVM emerging as the preferred model due to its rapid prediction times and superior performance in both binary and multi-class scenarios.

The study by Xu et al. proposed a novel 5-layer autoencoder (AE)-based model tailored for network anomaly detection tasks, aiming to address the limitations of existing state-of-the-art AE models [11]. Through extensive investigation of various performance indicators, the authors optimize model architecture and introduce a new data pre-processing methodology to mitigate data imbalance and reduce model bias. The proposed model utilizes an effective reconstruction error function to distinguish between normal and anomalous network traffic samples. Evaluation on the NSL-KDD dataset demonstrates superior performance compared to existing methods, achieving the highest accuracy and f1-score of 90.61% and 92.26%, respectively, in anomaly detection. The innovative approach enhanced feature learning and dimension reduction, leading to improved detection accuracy, overall model effectiveness and enhanced the network security.

III. Proposed work

The proposed model aims to develop a robust network anomaly detection system by exploring, preprocessing, and analyzing the NSL-KDD dataset, implementing deep autoencoder models, establishing anomaly detection mechanisms, and refining the system's performance through evaluation and refinement.

A. *Overall Architecture*

The work comprises several stages aimed at developing a robust network anomaly detection system. It begins with dataset exploration and preprocessing to understand its characteristics. After which data cleanliness is ensured. Next, a deep autoencoder model is designed and trained to learn latent representations of the input data, with performance evaluated using reconstruction error metric.

Following this, anomaly detection mechanisms are established based on reconstruction error thresholds, distinguishing between normal and anomalous data points. Concurrently, a clustering module is developed to partition data points into distinct clusters, each paired with a specialized autoencoder model as shown in Fig. 1. These paired models allow for tailored anomaly detection, with separate thresholds set based on cluster characteristics as shown in Fig. 2.

Finally, the complete system is evaluated using testing data, refining the model based on performance metrics to enhance its effectiveness in detecting network anomalies. This structured approach ensures a systematic development process, resulting in a reliable and comprehensive network anomaly detection framework.

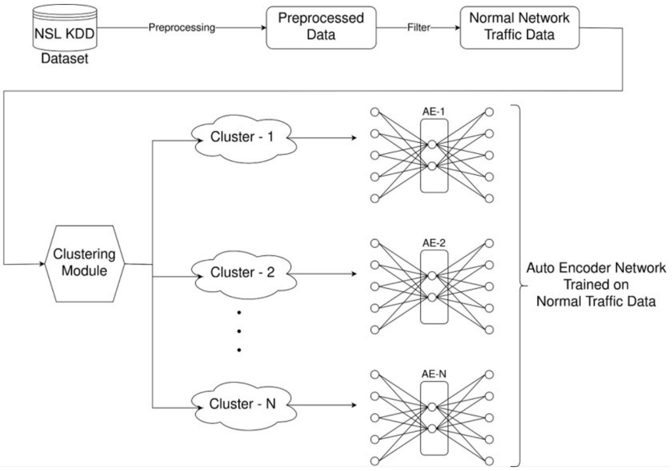

Fig. 1. Architecture Diagram (training)

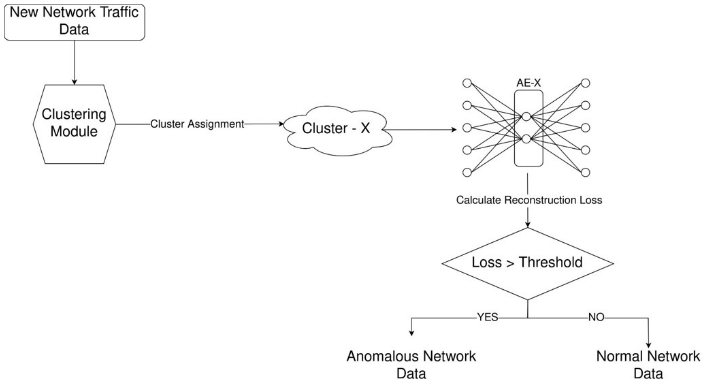

Fig. 2. Architecture Diagram (testing)

B. *Cluster Autoencoder Pair Algorithm*

The Algorithm 1 leverages the clustering information to guide the reconstruction process through an autoencoder, and subsequently uses the reconstruction loss to determine whether the input data point is anomalous or not. The threshold for each cluster allows for flexibility in identifying anomalies based on cluster-specific characteristics. The below is illustrated in Algorithm 1.

Algorithm 1 Autoencoder based Clustering Algorithm

```
Require: Network Traffic Data
Ensure: Anomaly Detection
Function DETECTANOMALY(datapoint)
cluster_no ← cluster_module(datapoint)
ae_output ← autoencoder_arr[cluster_no].predict(datapoint)
loss ← reconstruction_loss(ae_output, datapoint)
if loss > threshold_arr[cluster_no] then
    Print(”Anomalous Data”)
else
    Print(”Normal Data”)
end if
```

IV. Submodules Explanation

In the first stage, dataset exploration, analysis, and preprocessing are conducted on the NSL-KDD dataset [19] to ensure data quality. Subsequently, the second stage involves developing a deep autoencoder model trained on the preprocessed data for meaningful feature representation. Following this, the third stage implements anomaly detection mechanisms based on reconstruction error metrics to identify anomalies. Moving forward, the fourth stage utilizes clustering algorithms to group data points into distinct clusters, aiding in understanding underlying patterns. In the fifth stage, clustering is integrated with autoencoder models, allowing for the training of separate autoencoders for each cluster and setting individualized thresholds for anomaly detection. Finally, in the sixth stage, the system's performance is evaluated using testing data, with appropriate metrics employed to refine the model and enhance overall effectiveness based on evaluation results.

A. *Dataset exploration, Analysis and Preprocessing*

As part of implementation, exploring the NSL-KDD dataset involves scrutinizing its structure, identifying the number of features and samples, and assessing data types and distributions. Subsequent analysis and preprocessing encompass addressing missing values through imputation or removal, encoding categorical variables, and scaling numerical features to ensure uniformity in scale. Data cleaning steps, such as removing duplicates and rectifying inconsistencies, are executed to enhance data quality. Scaling techniques like standardization or normalization are then applied to numerical features to mitigate issues arising from varying scales. This comprehensive process ensures that the NSL-KDD dataset is adequately prepared for downstream tasks, facilitating more robust analysis and modeling endeavors.

B. *Autoencoder Model Development*

The deep autoencoder architecture was designed with multiple layers, including an input layer, several hidden layers for encoding, and symmetric layers for decoding, each employing ReLU activation functions to introduce nonlinearity. The autoencoder was implemented and trained on pre-processed data, undergoing steps such as data cleaning, scaling, and encoding. The model was evaluated using reconstruction error metrics, indicating its ability to effectively compress and reconstruct the input data, with lower reconstruction errors signifying successful learning of essential data features.

C. *Anomaly Detection Using Reconstruction Comparison*

For anomaly detection using reconstruction comparison, a thresholding mechanism based on reconstruction error was established to identify anomalies within the data. By analyzing the distribution of reconstruction errors, a suitable threshold value was determined to distinguish between normal data points and anomalies. Data points with reconstruction errors exceeding this threshold were classified as anomalies, enabling the effective detection of unusual patterns or outliers within the dataset.

D. *Clustering*

For clustering, a suitable algorithm was selected based on the dataset's characteristics and problem requirements. Parameters such as the number of clusters were fine-tuned through techniques like elbow method or silhouette analysis to achieve optimal performance. Subsequently, the data points were clustered into their respective bins or clusters using the chosen algorithm, enabling the identification of inherent patterns and groupings within the dataset for further analysis and interpretation.

E. *Cluster Autoencoder Pair*

In the integration of the Cluster Autoencoder Pair (CAEP), the clustering module was seamlessly incorporated with the autoencoder model to enhance anomaly detection capabilities. The training data was clustered, and separate autoencoders were then trained on data from each cluster. By setting individual thresholds for each autoencoder based on the reconstruction error metrics specific to their respective clusters, the system achieved a nuanced approach to anomaly detection, effectively capturing anomalies tailored to the characteristics of each cluster while minimizing false positives. This approach facilitated a more precise and adaptive anomaly detection mechanism, improving the overall robustness and accuracy of the system.

F. *Evaluation*

In the evaluation phase, the complete system was rigorously assessed using testing data to gauge its performance. Relevant metrics such as precision, recall, and F1-score were employed to quantitatively measure the system's effectiveness in anomaly detection. Based on the evaluation results, the model underwent refinement processes, which may include adjusting hyperparameters, fine-tuning thresholds, or revisiting feature selection strategies, to further optimize its performance. This iterative refinement cycle ensures the continuous improvement of the system's accuracy and reliability in detecting anomalies, thereby enhancing its practical utility and real-world applicability.

G. *Visualization of Clustering and Distance Analysis*

The next module includes distance analysis of normal and anamolus points. Initially, a KMeans model is trained on a dataset containing normal data points, facilitating the clustering of data into distinct groups. Subsequently, cluster labels are assigned to both normal and anomalous data instances, aiding in the characterization of their association with specific clusters. Following this, Euclidean distances between data points and their respective cluster centers are calculated, enabling a quantification of the conformity of each point to its assigned cluster. By visualizing the distributions of these distances for normal and anomalous data through histograms, the code enables a comparative analysis, with anomalous data points typically exhibiting larger distances from their cluster centers. FOr these visualizations t-SNE is used. This approach provides a robust framework for identifying outliers or anomalies within the dataset, offering a versatile solution applicable across various domains for effective anomaly detection.

H. *Anomaly Detection using Distance Metric*

Initially clustering is done. Once the clustering is done, we set the threshold and based on that we say if it is anamolus or not. In this refined anomaly detection methodology, the proposed work iteratively organizes the normal data points into clusters, extracting cluster points and calculating their distances to the corresponding cluster centers using pairwise distances. By segmenting the data in this manner, the algorithm gains a more granular understanding of the distribution of distances within each cluster, enabling a finer assessment of data point conformity. Similarly, for anomalous data, the code assigns cluster labels and computes the distances of these points to their respective cluster centers. Through this process, the algorithm not only discerns anomalies based on their distances from cluster centers but also captures the heterogeneity within each cluster. The visualization of these distributions through histograms elucidates the distinct patterns of normal and anomalous data, empowering analysts to pinpoint outliers with heightened precision. This refined approach, characterized by its focus on cluster-centric distance analysis, offers a robust framework for anomaly detection across diverse datasets, enhancing the sensitivity and specificity of anomaly detection algorithms in various application domains.

I. Autoencoder based Clustering

Autoencoder-based clustering is a method that combines unsupervised learning through autoencoders with clustering techniques to identify patterns and clusters within data. In this approach, an autoencoder, a type of artificial neural network, is trained to reconstruct input data with minimal error. By encoding input data into a lower-dimensional representation and then decoding it back to its original form, the autoencoder learns to capture meaningful features or representations of the data. After training, the encoder part of the autoencoder can be used to transform the data into a lower-dimensional space. Then, traditional clustering algorithms like KMeans can be applied to this encoded representation to partition the data into clusters. This method leverages the ability of autoencoders to capture complex patterns in high-dimensional data while also utilizing clustering algorithms to group similar data points together in a lower-dimensional space. Two different autoencoders are used to compare and validate the results.

V. Results and Discussion

The dataset after preprocessing was trained and tested using the autoencoder model. The model has about 90% accuracy. The model is also able to predict pretty well if a particular data is anomalous or not.

A. Dataset

The NSL-KDD dataset [19] is a widely used benchmark dataset in the field of cybersecurity and intrusion detection. It is an updated version of the original KDD Cup 99 dataset, which was created to evaluate intrusion detection systems for network security. The NSL-KDD dataset addresses some limitations of the KDD Cup 99 dataset, such as redundancy and lack of variety in attack types. It consists of network traffic data collected from a simulated environment, including both normal and various types of attack activities. The dataset is labeled with different attack categories, making it suitable for training and evaluating intrusion detection systems. The NSL-KDD dataset is commonly used to develop and test machine learning models for detecting and classifying network intrusions and cyberattacks.

B. Implementation Details

The Elbow method is a heuristic technique used to determine the optimal number of clusters in a dataset for clustering algorithms. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and selecting the ”elbow” point where the rate of decrease in WCSS sharply decreases, indicating the appropriate number of clusters to use for partitioning the data. The graph is shown in Figure 3.

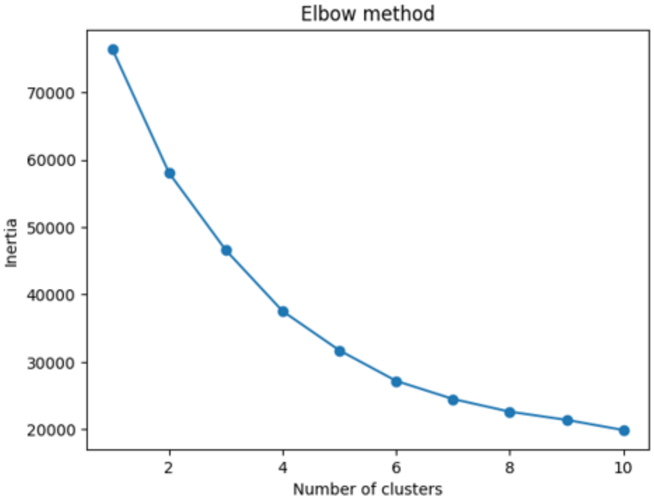

Fig. 3. Number of clusters evaluation using Elbow method

The Silhouette score is a metric used to evaluate the quality of clustering in a dataset. It quantifies how similar an object is to its own cluster compared to other clusters, with scores ranging from -1 to 1, where a higher score indicates better clustering structure. The corresponding Score for each cluster is shown in Table I. On taking the average of the scores we get approximately 0.411 so we conclude that number of clusters to be 6.

TABLE I
Silhouette Scores

No. of Clusters	Silhouette Score
2	0.379949
3	0.4011662
4	0.4061014
5	0.3665990
6	0.406159
7	0.434782
8	0.445773

The no of points for every cluster is shown in Table II. Cluster 1 has the most no of points with over 20000 while Cluster 0 has the least points with 6970 points.

TABLE II
CLUSTER DISTRIBUTION

Clusters No.	No. of Points
0	6970
1	20022
2	16433
3	2897
4	11303
5	9429

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a dimensionality reduction technique commonly used for visualizing high-dimensional data in lower-dimensional space. It preserves local structure by modeling similarity between data points, often revealing clusters or patterns that may be hidden in the original data. For the dataset t-SNE is applied and is shown in Figure 4.

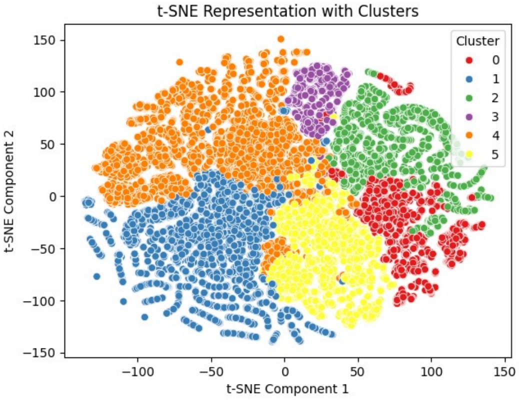

Fig. 4. Clustering visualization using t-SNE

PCA (Principal Component Analysis) is a technique used for dimensionality reduction by transforming high-dimensional data into a lower-dimensional space while preserving the most important information. It achieves this by identifying the principal components, which are orthogonal vectors that capture the maximum variance in the data. The PCA for the dataset is shown below in Figure 5.

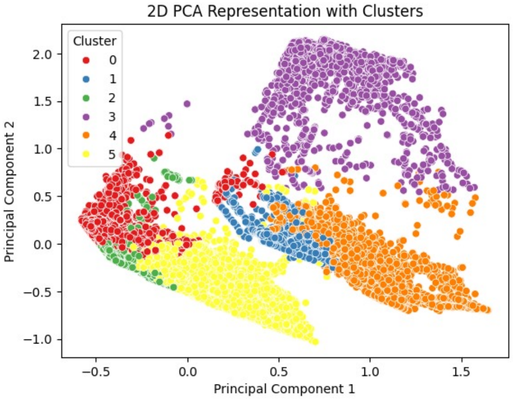

Fig. 5. Clustering visualization using PCA

The autoencoder implemented here is a feedforward neural network consisting of an encoder and a decoder. The encoder compresses the input data into a lower-dimensional latent space representation through two hidden layers with 32 and 16 neurons, respectively, utilizing the ReLU activation function to introduce nonlinearity. Subsequently, the decoder aims to reconstruct the original input data from this compressed representation through two hidden layers with 32 neurons each, employing the ReLU activation function. The final layer uses the sigmoid activation function to ensure output values are in the range [0, 1]. The autoencoder is trained using the Adam optimizer and mean squared error loss function, with accuracy as a metric. The autoencoder model is done with One Hot Encoding and the results are shown in Figure 6.

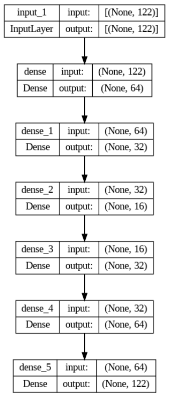
input_1 | input: | [(None, 122)]
InputLayer | output: | [(None, 122)]
dense | input: | (None, 122)
Dense | output: | (None, 64)
dense_1 | input: | (None, 64)
Dense | output: | (None, 32)
dense_2 | input: | (None, 32)
Dense | output: | (None, 16)
dense_3 | input: | (None, 16)
Dense | output: | (None, 32)
dense_4 | input: | (None, 32)
Dense | output: | (None, 64)
dense_5 | input: | (None, 64)
Dense | output: | (None, 122)

Fig. 6. One Hot Encoding - Autoencoder model

The same is implemented using Label Encoder and the results are shown in Figure 7. In the context of autoencoders, label encoding was typically applied to categorical variables before feeding them into the network. This process involved converting categorical variables into numerical labels, assigning a unique integer to each category.

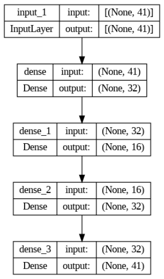
input_1 | input: | [(None, 41)]
InputLayer | output: | [(None, 41)]
dense | input: | (None, 41)
Dense | output: | (None, 32)
dense_1 | input: | (None, 32)
Dense | output: | (None, 16)
dense_2 | input: | (None, 16)
Dense | output: | (None, 32)
dense_3 | input: | (None, 32)
Dense | output: | (None, 41)

Fig. 7. Label Encoding - Autoencoder model

t-SNE representation is used like before to show both the anomalous and normal points of the 6 clusters. This gives understanding as to how within the cluster normal and anomalous points are. The above is illustrated in Figure 8

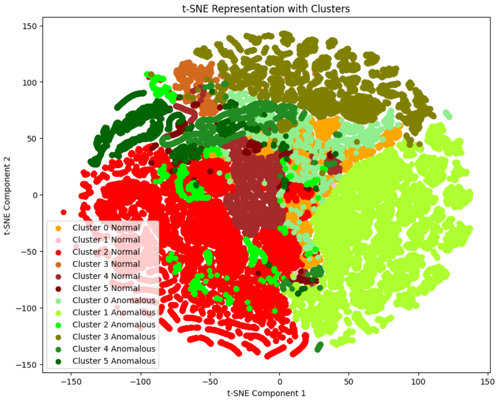

Fig. 8. t-SNE representation of both normal and anomalous points

In the study, we examined the impact of different encoding techniques on the overall performance of the system. Initially, the dataset was encoded using one-hot encoding, resulting in a feature set of 122 dimensions. Subsequently, an autoencoder model was trained on this one-hot encoded data, achieving an accuracy of 88%. However, upon further evaluation, it was observed that this high-dimensional representation introduced computational overhead without significantly improving performance. The above is shown in Figure 9

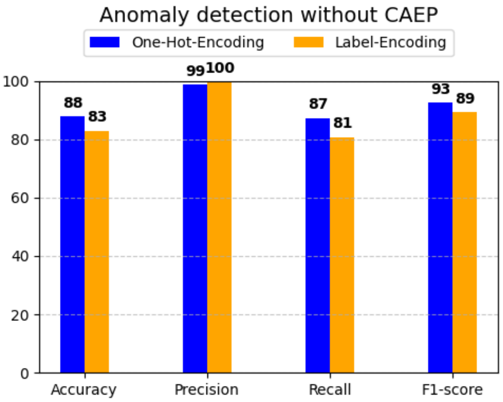

Fig. 9. Anomaly detection without CAEP - Performance

In contrast, employing label encoding led to a feature set of only 41 dimensions, with the autoencoder achieving a slightly lower accuracy of 83%. Despite the lower dimensionality and marginally reduced accuracy of the autoencoder, the label-encoded representation proved to be more efficient in terms of computational resources. The above is shown in Figure 9

This highlights the importance of considering not only individual model performance but also the holistic impact of encoding techniques on the overall system architecture and performance. Additionally, it underscores the potential for more computationally efficient representations, such as label encoding, to achieve comparable results while reducing computational overhead, thereby improving the scalability and practicality of the system. Further investigation into the trade-offs between dimensionality reduction, model accuracy, and computational efficiency could provide valuable insights for optimizing the system's encoding strategy in future iterations. The accuracies without using CAEP is shown in Figure 9 . As shown in Figure 10, the accuracies have improved both for one-hot encoding and label-encoding after using CAEP. Figure 11 and Figure 12 shows the corresponding confusion matrix.

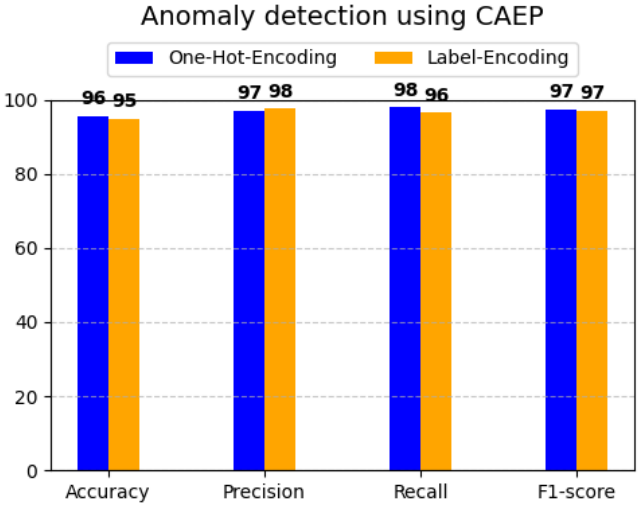

Fig. 10. Cluster-Autoencoder-Pair - Performance

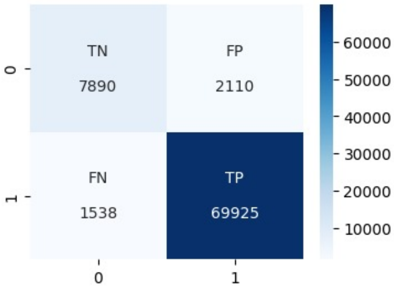

Fig. 11. Confusion Matrix using CAEP (one-hot-encoded)

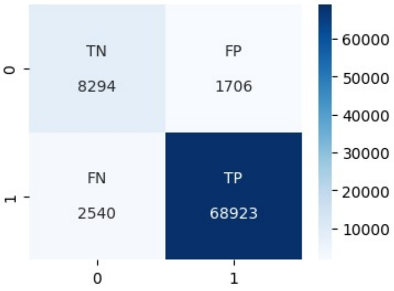

Fig. 12. Confusion Matrix using CAEP (label-encoded)

The Figure 13 shows anomaly detection using KMeans clustering by training the model on normal data, predicting cluster labels for both normal and anomalous data, and then comparing the distances between data points and cluster centers to distinguish between normal and anomalous instances.

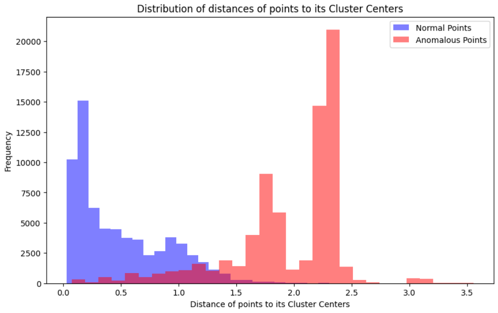

Fig. 13. Distance of points to it's Cluster Centers

The proposed work generates six histograms, each illustrating the distribution of distances between data points and their respective cluster centers for both normal and anomalous instances within each cluster. These histograms provide a detailed view of how well-separated the clusters are from their centers and how anomalous points deviate from their respective cluster distributions. Figure 14, Figure 15, Figure 16, Figure 17, Figure 18, and Figure 19 depict these distributions for each cluster, offering insights into the clustering effectiveness and the characteristics of anomalies within each cluster. The overall accuracy on test data achieved with the test data is 89%.

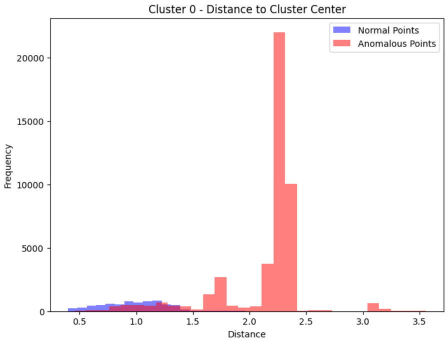

Fig. 14. Cluster 0 - Distance to Cluster Center

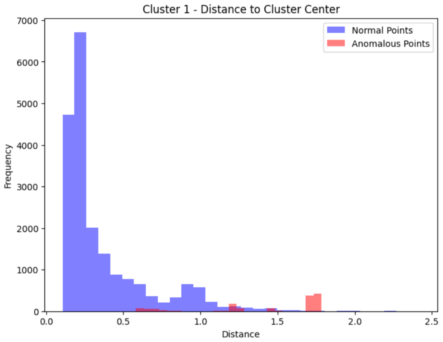

Fig. 15. Cluster 1 - Distance to Cluster Center

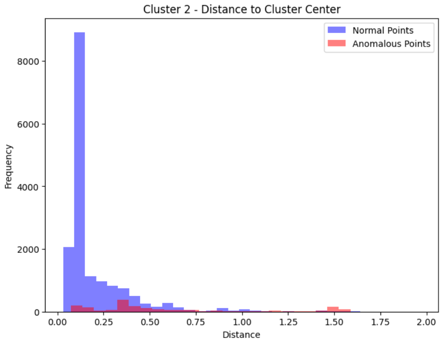

Fig. 16. Cluster 2 - Distance to Cluster Center

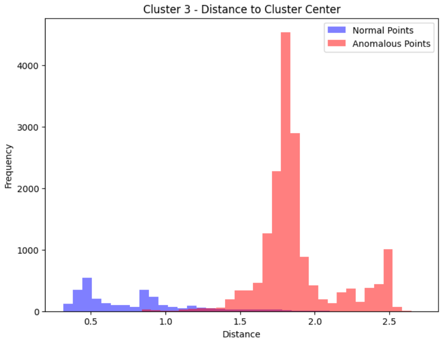

Fig. 17. Cluster 3 - Distance to Cluster Center

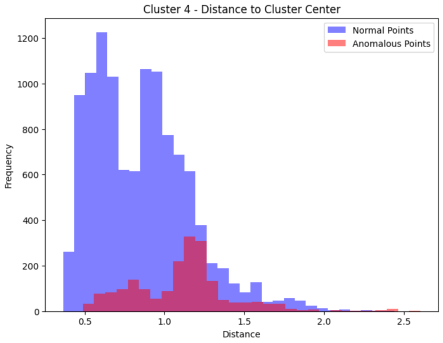

Fig. 18. Cluster 4 - Distance to Cluster Center

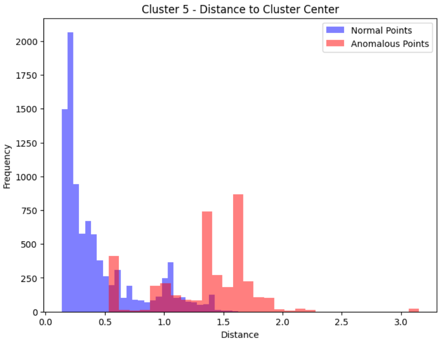

Fig. 19. Cluster 5 - Distance to Cluster Center

The Figure 20 represents the performance that is achieved for anomaly detection while using distance metric. An accuracy of 91% is achieved through this method.

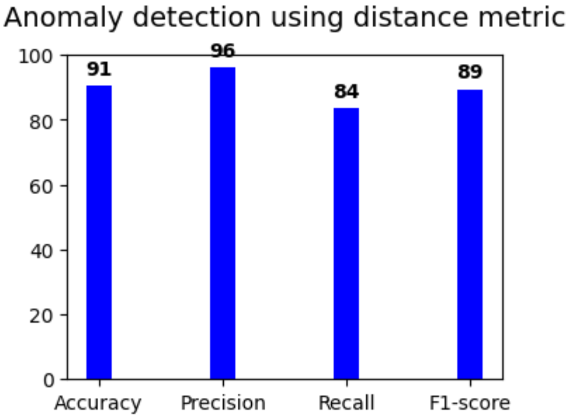

Fig. 20. Anomaly Detection using Distance Metric - Performance

Autoencoder based clustering is done here. Autoencoder 1 is the initial autoencoder used. Figure 21 shows the autoencoder 1 which takes in an input level of 41 parameters and then further reduces it to 32 layers and then 16 layers to finally 8 layers.

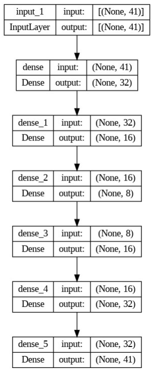

Fig. 21. Auto-encoder 1 trained for Auto-encoder based clustering

For autoencoder based clustering another autoencoder model is tested. This is autoencoder 2. Figure 22 shows the autoencoder 2 which takes in an input level of 41 parameters and then further reduces it to 32 layers and then 16 layers.

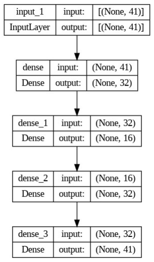

Fig. 22. Auto-encoder 2 trained for Auto-encoder based clustering

The accuracy while using Autoencoder-1 as shown in Figure 23 is 84. However, The accuracy while using Autoencoder-2 as shown in Figure 23 is 78. This clearly shows that Autoencoder-1 provides better performance when compared to Autoencoder-2.

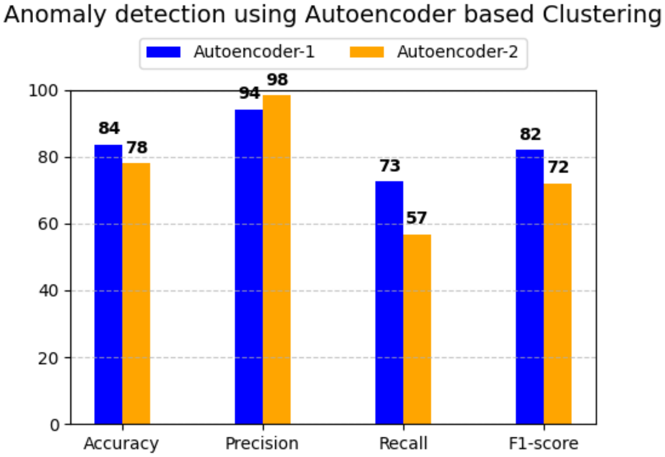

Fig. 23. Autoencoder based Clustering - Performance

C. *Ablation Study*

Upon developing the CAEP system, integrating the autoencoder models with the clustering module and testing them on unseen data, it was revealed that the CAEP system trained on the one-hot-encoded data scored an accuracy of 96% where as the CAEP system trained on label-encoded data scored an accuracy of 95%. It is to be noted that incorporating the CAEP system has brought a 8% increase in accuracy with the one-hot-encoded data. however, surprisingly when tested with the label-encoded data, the increase in performance was more significant with an 12% increase in accuracy. This unexpected result suggests that while one-hot encoding initially produced a higher accuracy with more features, the additional complexity did not significantly contribute to the system's overall performance. In contrast, the label-encoded representation, despite its lower dimensionality and slightly reduced accuracy at the autoencoder level, proved to be equally effective in the context of the complete system. The accuracy achieved through anomaly detection using distance metric is 91%.

Of the different methods explored the novel proposed work CAEP showed highest accuracy of 96% when used with One-Hot encoding compared to 94% when used with Label Encoding. Autoencoder based clustering achieved just 84% and 78% accuracy for the 2 autoencoders tested. And the distance metric just achieved 91%. Hence the proposed CAEP method is chosen to be the best method of the other 3 methods explored.

D. Comparison with Existing Work

In comparison with previous work, ARCADE achieved F1-scores of 91.95% and 93.19% for HTTP DoS and DDoS attacks, respectively. Notably, our proposed model utilizing the Clustered AutoEncoder Pair method attained a higher accuracy rate of 96% is tabulated in Table III. Furthermore, our model demonstrates significant improvements in various aspects, including accuracy, model complexity, and detection speed, despite being 20 times smaller than the baselines used in previous studies. This highlights the efficacy and efficiency of our approach in enhancing cybersecurity measures, particularly in the detection of network attacks. The above results are shown in Table III

TABLE III
F1-Score Comparison

Work	F1-Score
ARCADE DOS by Lunardi et al.[1]	91.95%
ARCADE DDOS by Lunardi et al.[1]	93.19%
Proposed CAEP	**97%**

The Table IV presents a comparative analysis of various anomaly detection methods, showcasing their performance in terms of accuracy, precision, recall, and F1-score. Notable methods include NDAE by Shone et al., DBSCAN by Ursul et al., G-CNN-AE by Khan et al., AE-SVM by Khan et al., AE by Wen et al., Distance Metric based Anomaly Detection, AE Based Clustering, and the proposed CAEP method.

NDAE by Shone et al. achieves an accuracy of 89.22%, precision of 92.97%, recall of 89.22%, and F1-score of 90.76%. DBSCAN by Ursul et al. demonstrates precision, recall, and F1-score of 92%, 99%, and 95% respectively, with no accuracy provided. G-CNN-AE by Khan et al. shows promising results with accuracy, precision, recall, and F1-score of 93.2%, 97.3%, 96%, and 96.7% respectively. AE-SVM by Khan et al. yields lower accuracy (75%) and precision (68%) but moderate recall (90%) and F1-score (78%). AE by Wen et al. exhibits an accuracy of 90.61%, precision of 98.43%, recall of 86.83%, and F1-score of 92.26%.

Additionally, two baseline methods, Distance Metric based Anomaly Detection and AE Based Clustering, are included in the comparison, showing consistent performance across accuracy, precision, recall, and F1-score metrics.

Finally, the proposed CAEP method stands out with an impressive accuracy of 96%, precision of 95%, recall of 96%, and F1-score of 95%. This method leverages clustered autoencoder pairs for anomaly detection, suggesting promising potential for improved anomaly detection accuracy and robustness compared to existing approaches.

TABLE IV
Comparison with Previous Work

Work	Accuracy	Precision	Recall	F1-Score
NDAE by Shone et al. [9]	89.22%	92.97%	89.22%	90.76%
DBSCAN by Ursul et al. [6]	-	92%	99%	95%
G-CNN-AE by Khan et al. [4]	93.2%	97.3%	96%	96.7%
AE-SVM by Khan et al. [10]	75%	90%	68%	78%
AE by Wen et al. [11]	90.61%	86.83%	98.43%	92.26%
Distance Metric based Anomaly Detection	91%	91%	91%	91%
AE-1 Based Clustering	84%	86%	84%	84%
AE-2 Based Clustering	78%	84%	78%	77%
Proposed CAEP	**96%**	**98%**	**98%**	**97%**

VI. Conclusion

A. Conclusion

The escalating sophistication of cyber threats has underscored the inadequacy of conventional security measures, necessitating the adoption of advanced and adaptable solutions. This project proposes a novel approach to cybersecurity through the development and integration of intelligent systems capable of recognizing anomalous network activity. By leveraging deep learning techniques, particularly through the implementation of autoencoder models and clustering algorithms, the system demonstrates promising capabilities in early threat detection and risk mitigation. The successful execution of modules such as dataset exploration, autoencoder model development, anomaly detection, and clustering underscores the feasibility and efficacy of this approach. Through rigorous evaluation and refinement, the system exhibits potential for enhancing network security by proactively identifying and addressing vulnerabilities before they escalate into significant security incidents.

B. Future Work

In order to further enhance the system's performance and reduce false positives (FP) and false negatives (FN), future work could involve conducting a comparative study of other related approaches for anomaly detection, Exploring variations in clustering algorithms and autoencoder architectures could offer insights into optimizing the system for improved anomaly detection accuracy. Additionally, investigating ensemble methods that combine multiple anomaly detection models, could potentially mitigate FP and FN rates by leveraging diverse perspectives and enhancing the system's overall resilience to false alarms and missed detections. Furthermore, fine-tuning thresholds and refining anomaly detection mechanisms based on real-world feedback and domain-specific knowledge could help tailor the system to specific use cases and environments, thereby maximizing its effectiveness in mitigating cyber threats while minimizing disruptive false alarms. This iterative approach to research and development aims to continuously refine and evolve the system towards achieving a balance between proactive threat detection and minimizing false positives and negatives.

References

[1] W. T. Lunardi, M. A. Lopez and J. -P. Giacalone, "ARCADE: Adversarially Regularized Convolutional Autoencoder for Network Anomaly Detection," in IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1305-1318, June 2023, doi: 10.1109/TNSM.2022.3229706.

[2] C. Aytekin, X. Ni, F. Cricri and E. Aksu, "Clustering and Unsupervised Anomaly Detection with L2 Normalized Deep Auto-Encoder Representations," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-6, doi: 10.1109/IJCNN.2018.8489068.

[3] C. Maudoux and S. Boumerdassi, "Network Anomalies Detection by Unsupervised Activity Deviations Extraction," 2022 Global Information Infrastructure and Networking Symposium (GIIS), Argostoli, Greece, 2022, pp. 1-5, doi: 10.1109/GIIS56506.2022.9937022.

[4] C. Park, J. Lee, Y. Kim, J. -G. Park, H. Kim and D. Hong, "An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks," in IEEE Internet of Things Journal, vol. 10, no. 3, pp. 2330-2345, 1 Feb.1, 2023, doi: 10.1109/JIOT.2022.3211346.

[5] H. Benaddi, K. Ibrahimi, A. Benslimane, M. Jouhari and J. Qadir, "Robust Enhancement of Intrusion Detection Systems Using Deep Reinforcement Learning and Stochastic Game," in IEEE Transactions on Vehicular Technology, vol. 71, no. 10, pp. 11089-11102, Oct. 2022, doi: 10.1109/TVT.2022.3186834.

[6] I. Ursul and A. Pereymybida, "Unsupervised Detection of Anomalous Running Patterns Using Cluster Analysis," 2023 IEEE 13th International Conference on Electronics and Information Technologies (ELIT), Lviv, Ukraine, 2023, pp. 148-152, doi: 10.1109/ELIT61488.2023.10310751.

[7] M. A. Ambusaidi, X. He, P. Nanda and Z. Tan, "Building an Intrusion Detection System Using a Filter-Based Feature Selection Algorithm," in IEEE Transactions on Computers, vol. 65, no. 10, pp. 2986-2998, 1 Oct. 2016, doi: 10.1109/TC.2016.2519914.

[8] N. Ravi and S. M. Shalinie, "Semisupervised-Learning-Based Security to Detect and Mitigate Intrusions in IoT Network," in IEEE Internet of Things Journal, vol. 7, no. 11, pp. 11041-11052, Nov. 2020, doi: 10.1109/JIOT.2020.2993410.

[9] N. Shone, T. N. Ngoc, V. D. Phai and Q. Shi, "A Deep Learning Approach to Network Intrusion Detection," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41-50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[10] S. S. Khan and A. B. Mailewa, "Detecting Network Transmission Anomalies using Autoencoders-SVM Neural Network on Multi-class NSL-KDD Dataset," 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2023, pp. 0835-0843, doi: 10.1109/CCWC57344.2023.10099056.

[11] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei and F. Sabrina, "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset," in IEEE Access, vol. 9, pp. 140136-140146, 2021, doi: 10.1109/ACCESS.2021.3116612.

[12] Z. Xiao, Q. Yan, and Y. Amit, "Do we really need to learn representations from in-domain data for outlier detection?" 2021, arXiv:2105.09270.

[13] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, 2021, Art. no. e4150.

[14] Y. Yu, J. Long, and Z. Cai, "Network intrusion detection through stacking dilated convolutional autoencoders," Security and Communication Networks, vol. 2017, Nov. 2017, Art. no. 4184196.

[15] Y. B. Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-generation Internet of Things (IoT): Opportunities, challenges, and solutions," Sensors, vol. 21, no. 4, p. 1174, Feb. 2021.

[16] W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei and F. Sabrina, "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset," in IEEE Access, vol. 9, pp. 140136-140146, 2021, doi: 10.1109/ACCESS.2021.3116612.

[17] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in Proc. Int. Conf. Intell. Security Informat., 2017, pp. 43–48.

[18] T. Truong-Huu et al., "An empirical study on unsupervised network anomaly detection using generative adversarial networks," in Proc. 1st ACM Workshop Security Privacy Artif. Intell., 2020, pp. 20–29.

[19] https://www.unb.ca/cic/datasets/index.html