

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

IT5811 - A Project-II Report

Submitted by

Krishnaa S 2020506045

Jawahar A S 2020506035

Thamizharasi M 2020506102

Under the supervision of

Dr. P. Kola Sujatha

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY



DEPARTMENT OF INFORMATION TECHNOLOGY

MADRAS INSTITUTE OF TECHNOLOGY CAMPUS

ANNA UNIVERSITY, CHENNAI – 600044

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project report titled “**CLUSTERS IN CHAOS: A DEEP UNSUPERVISED LEARNING PARADIGM FOR NETWORK ANOMALY DETECTION**” is the bonafide work of Krishnaa S (2020506045), Jawahar A S (2020506035), Thamizharasi M (2020506102) who carried out the project work under my supervision.

Signature

Dr. M. R. Sumalatha

HEAD OF THE DEPARTMENT

Professor

Department of Information Technology

MIT Campus, Anna University

Chennai – 600044

Signature

Dr. P. Kola Sujatha

SUPERVISOR

Associate Professor

Department of Information Technology

MIT Campus, Anna University

Chennai – 600044

ACKNOWLEDGEMENT

It is essential to mention the names of the people, whose guidance and encouragement made us accomplish this project.

We express our thankfulness to our project supervisor **Dr.P. Kola Sujatha**, Department of Information Technology, MIT Campus, for providing invaluable support and assistance with encouragement which aided to complete this project.

We are very thankful to the panel members **Dr. P. AnandhaKumar**, and **Dr. M. Hemalatha**, Department of Information Technology, MIT Campus for their invaluable feedback in reviews.

Our sincere thanks to **Dr. M. R. Sumalatha**, Head of the Department of Information Technology, MIT Campus for catering all our needs giving out limitless support throughout the project phase.

We express our gratitude and sincere thanks to our respected Dean of MIT Campus, **Prof. Ravichandran Kandaswamy**, for providing excellent computing facilities throughout the project.

Krishnaa S 2020506045

Jawahar A S 2020506035

Thamizharasi M 2020506102

ABSTRACT

Traditional security measures are inadequate against the escalating sophistication of cyber threats, leading to increased disruptions and financial losses. To address this, intelligent systems capable of recognizing unusual network activity are essential. This research focuses on implementing such systems, emphasizing analysis of the NSL KDD dataset. A deep unsupervised learning model is developed, augmented by a novel approach using Clustered Autoencoders to enhance anomaly detection. This innovative technique groups similar normal data points and trains individual autoencoders for each cluster, improving accuracy and versatility in handling diverse network traffic scenarios. These advancements signify a significant leap forward in anomaly detection, addressing limitations and showcasing practical applicability through empirical analysis. By leveraging intelligent systems and cutting-edge techniques, organizations can better identify and mitigate cyber threats, safeguarding critical assets and infrastructure in today's dynamic threat landscape.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	ix
1	INTRODUCTION	
	1.1 Overview	1
	1.2 Research Challenges	1
	1.3 Objective	2
	1.4 Scope of the Project	2
	1.5 Contribution	3
	1.6 Network Anomaly	3
	1.7 Deep Learning	4
	1.8 Clustering	5
	1.9 Auto-Encoders	5
	1.10 NSL-KDD Dataset	6
	1.11 Unsupervised Learning	6
	1.12 Organization of the Thesis	7
2	LITERATURE SURVEY	
	2.1 Convolutional Autoencoder Approach	8
	2.2 Normalized Deep Autoencoder Approach	8
	2.3 Sector based Unsupervised Clustering Approach	9
	2.4 Synthetic Data Augmentation Approach	9
	2.5 Deep Reinforcement Learning Approach	9
	2.6 Cluster based Approach	10
	2.7 Mutual Information Feature Selection Approach	10

	2.8 SDRK Machine Learning Approach	11
	2.9 Non- Symmetric Deep Autoencoder Approach	11
	2.10 Support Vector Machine Approach	11
	2.11 Summary of the Literature Survey	12
3	SYSTEM ARCHITECTURE AND DESIGN	
	3.1 System Architecture	13
	3.2 Understanding Network Anomalies	15
	3.3 Anomaly Detection Algorithms	15
	3.4 Balancing Accessibility and Security	15
4	ALGORITHM DEVELOPMENT AND IMPLEMENTATION	
	4.1 Algorithm	17
	4.2 Implementation	18
	4.3 Dataset Exploration, Analysis and Preprocessing	18
	4.4 Autoencoder Model Development	19
	4.5 Anomaly Detection Using Reconstruction	
	Comparison	19
	4.6 Clustering	19
	4.7 Cluster- Autoencoder Pair	20
	4.8 Evaluation	20
	4.9 Anomaly Detection Via Distance Metric	21
	4.10 Visualization of Distance Analysis	21
	4.11 Auto-encoder based Clustering	22

5	RESULTS AND DISCUSSIONS	
	5.1 Dataset	25
	5.2 Implementation Details	25
	5.3 Ablation Study	36
	5.4 Comparison with Existing Work	39
6	CONCLUSIONS AND FUTURE WORKS	
	6.1 Conclusions	40
	6.2 Future Works	40
	REFERENCES	42

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
3.1	System Architecture (Training)	14
3.2	System Architecture (Testing)	14
5.1	Number of clusters evaluation using Elbow method	24
5.2	Number of clusters evaluation using Silhouette score	24
5.3	Cluster Distribution	25
5.4	Clustering visualization using t-SNE	25
5.5	Clustering visualization using PCA	26
5.6	One Hot Encoding - Autoencoder model	27
5.7	Label Encoding - Autoencoder model	28
5.8	t-SNE representation of both normal and anomalous points	28
5.9	Distance of points to it's Cluster Centers	29
5.10	Cluster 0 - Distance to Cluster Center	30
5.11	Cluster 1 - Distance to Cluster Center	30
5.12	Cluster 2 - Distance to Cluster Center	30
5.13	Cluster 3 - Distance to Cluster Center	31
5.14	Cluster 4 - Distance to Cluster Center	31
5.15	Cluster 5 - Distance to Cluster Center	31
5.16	Performance of Distance metric based Anomaly detection	32
5.17	Auto-encoder 1	33
5.18	Auto-encoder 1 performance	33
5.19	Auto-encoder 2	34

5.20	Auto-encoder 2 performance	34
5.21	Using one-hot-encoding without CAEP	35
5.22	Using label encoding without CAEP	36
5.23	Using one-hot-encoding with CAEP	37
5.24	Using label encoding with CAEP	37
5.25	Confusion Matrix using CAEP (one-hot-encoded)	38
5.26	Confusion Matrix using CAEP (label-encoded)	38

LIST OF ABBREVIATIONS

ABC	Auto-Encoder Based Clustering
ARCADE	Adversarial Regularized Convolutional Autoencoder for unsupervised network anomaly detection
CAEP	Cluster-Autoencoder-Pair
CNN	Convolutional Neural Networks
DAE	Deep Autoencoder
FN	False Negatives
FP	False Positives
FMIFS	Flexible Mutual Information Feature Selection
GAN	Generative Adversarial Network
IDS	Intrusion Detection System
KDD	Knowledge Discovery in Databases
NDAE	Non-Symmetric Deep Autoencoder
NIDS	Network Intrusion Detection System
NSL	University of New South Wales
PCA	Principal Component Analysis
RNN	Recurrent Neural Networks
SDA	Synthetic Data Augmentation
SVM	Support Vector Machine
t-SNE	t-Distributed Stochastic Neighbor Embedding
WCSS	Within-Cluster Sum of Squares

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The landscape of cyber threats is evolving rapidly, surpassing the capabilities of traditional security measures to adequately protect digital infrastructure. As cyberattacks grow more sophisticated and diverse in nature, it has become evident that conventional security protocols are no longer sufficient to defend against them effectively. There is an urgent requirement for innovative and adaptable solutions that can detect and counter these evolving threats in a proactive manner. In response to this need, intelligent systems have emerged as a promising approach to bolster cybersecurity defences. These systems are equipped with the capability to monitor network activity in real-time and identify any deviations from normal behavior. What sets these systems apart is their utilization of deep learning techniques, which enable them to process and analyze vast volumes of network data with unprecedented speed and accuracy. By leveraging deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), these intelligent systems can discern subtle patterns and anomalies within the network traffic. This ability to detect anomalies in real-time allows organizations to take proactive measures to mitigate potential security threats before they escalate into serious incidents. In essence, intelligent systems empowered by deep learning represent a paradigm shift in cybersecurity. They offer a proactive and adaptive approach to threat detection and mitigation, enabling organizations to stay one step ahead of cyber adversaries in an increasingly dynamic and complex threat landscape.

1.2 RESEARCH CHALLENGES

The complexity of cyber threats presents a significant challenge in the realm of cybersecurity. As adversaries continuously innovate and adapt their

tactics, traditional detection methods struggle to keep pace. This complexity extends to the diverse range of attack vectors, including malware, phishing, and denial-of-service attacks, each requiring unique detection and mitigation strategies. Additionally, the sheer volume of network data generated in today's interconnected world adds another layer of complexity. Analyzing this data in real-time to identify anomalies indicative of potential threats requires robust computational infrastructure and advanced data processing techniques. Moreover, interpreting these anomalies accurately poses a challenge, as distinguishing between benign fluctuations and genuine security breaches demands a deep understanding of network behavior and threat landscape dynamics.

1.3 OBJECTIVE

The primary objective of this research is to pioneer intelligent systems capable of proactively detecting and mitigating cyber threats through the application of deep learning techniques to analyze network traffic data. By delving into the effectiveness of deep learning models, the study seeks to uncover patterns and anomalies indicative of potential security breaches within network communications. Through the development of innovative algorithms and methodologies, the aim is to enable real-time analysis of network data, empowering organizations to swiftly identify and respond to emerging threats. Ultimately, the research endeavors to evaluate the performance of these intelligent systems in bolstering network security by mitigating cyber risks before they escalate into serious security incidents.

1.4 SCOPE OF THE PROJECT

The scope of this project encompasses an in-depth exploration of deep learning algorithms, specifically auto-encoders(AE) , with the aim of leveraging their capabilities for anomaly detection in network traffic data. Through the development of prototypes, the project will focus on creating

intelligent systems capable of analyzing vast amounts of network data in real-time. Experimentation and simulation will be conducted to assess the effectiveness and scalability of these solutions in detecting and mitigating cyber threats efficiently. Additionally, collaboration with industry partners will be sought to validate the practical applicability of the developed systems in real-world scenarios, ensuring that the research outcomes have tangible benefits for enhancing network security. By addressing these objectives, the project aims to contribute significantly to the advancement of proactive cybersecurity measures through the utilization of cutting-edge deep learning techniques.

1.5 CONTRIBUTION

This research makes significant contributions to the field of cybersecurity by introducing intelligent systems that elevate the standard of threat detection and mitigation. By leveraging deep learning techniques, particularly for anomaly detection in network traffic, the study offers valuable insights into the effectiveness of advanced algorithms in enhancing network security. Moreover, the research addresses critical challenges associated with implementing and deploying intelligent cybersecurity solutions, thereby paving the way for more robust and adaptable defence mechanisms. By providing practical recommendations based on empirical findings, the study contributes to enhancing network security and resilience against the constantly evolving landscape of cyber threats. Ultimately, this research not only expands the knowledge base in cybersecurity but also provides tangible solutions that can be applied in real-world scenarios to bolster defence strategies and safeguard digital infrastructure against emerging threats.

1.6 NETWORK ANOMALY

Network anomaly refers to any deviation from normal patterns or behaviours in network traffic, signalling potential security threats or operational issues. These anomalies can manifest as unusual spikes in data traffic, unexpected

protocol usage, unauthorized access attempts, or suspicious communication patterns. Detecting and analysing network anomalies is critical for maintaining the integrity and security of computer networks, as they may indicate malicious activities such as cyberattacks, intrusions, or data breaches. Traditional methods of anomaly detection rely on rule-based approaches or statistical analysis, but with the increasing complexity and sophistication of cyber threats, there is a growing need for more advanced techniques. Machine learning, particularly deep learning algorithms, has emerged as a promising approach for detecting network anomalies, as it can automatically learn and adapt to evolving patterns in network data. By effectively identifying and responding to network anomalies, organizations can enhance their cybersecurity posture and mitigate potential risks to their network infrastructure and sensitive data.

1.7 DEEP LEARNING

Deep learning is a subset of machine learning that leverages artificial neural networks with multiple layers to automatically learn intricate patterns and representations from data. Unlike traditional machine learning approaches, which often require manual feature engineering, deep learning algorithms can autonomously extract relevant features directly from raw data, making them highly effective for tasks such as image recognition, natural language processing, and speech recognition. The depth and complexity of these neural networks enable them to capture hierarchical representations of data, allowing for more nuanced understanding and interpretation of complex relationships. Deep learning has revolutionized various fields, including computer vision, healthcare, finance, and cybersecurity, by achieving state-of-the-art performance on a wide range of tasks. In cybersecurity, deep learning techniques are increasingly used for anomaly detection, intrusion detection, malware analysis, and threat intelligence, offering advanced capabilities for identifying and mitigating security threats in real-time.

1.8 CLUSTERING

Clustering is a fundamental technique in machine learning and data analysis that involves grouping similar data points together based on certain features or characteristics. The goal of clustering is to partition a dataset into subsets, or clusters, such that data points within the same cluster are more similar to each other than to those in other clusters. Common clustering algorithms include K-means, hierarchical clustering, and DBSCAN. Clustering is widely used in various domains such as customer segmentation, image segmentation, anomaly detection, and recommendation systems. It helps identify hidden patterns and structures within data, facilitating insights and decision-making processes. Clustering algorithms rely on distance metrics or density-based approaches to measure similarity between data points and determine cluster membership. However, choosing the appropriate clustering algorithm and determining the optimal number of clusters remain ongoing challenges in clustering analysis.

1.9 AUTO-ENCODERS

Autoencoders are a type of artificial neural network used for unsupervised learning that aims to learn efficient representations of input data. They consist of an encoder network that compresses the input data into a lower-dimensional representation, known as the latent space, and a decoder network that reconstructs the original input from this representation. By minimizing the reconstruction error between the input and the output, autoencoders learn to capture the most salient features of the data. They are particularly useful for dimensionality reduction, feature learning, and data denoising tasks. Additionally, autoencoders can be adapted for various applications, including image reconstruction, anomaly detection, and generative modeling. Variants of autoencoders, such as convolutional autoencoders and variational autoencoders, have been developed

to handle different types of data and address specific challenges, making them versatile tools in the field of deep learning.

1.10 NSL-KDD DATASET

The NSL-KDD (University of New South Wales - Knowledge Discovery in Databases) dataset is a benchmark dataset widely used for evaluating intrusion detection systems (IDS) and anomaly detection algorithms in network security. It is an improved version of the original KDD Cup 99 dataset, addressing some of its limitations such as redundancy and bias. The NSL-KDD dataset contains network traffic data generated from a simulated environment, including various types of attacks and normal activities. It provides a diverse range of features, such as protocol types, service types, and connection attributes, making it suitable for training and testing machine learning models for intrusion detection. The dataset is annotated with labels indicating different types of attacks, allowing researchers to evaluate the performance of their algorithms in detecting and classifying malicious activities accurately. Due to its realistic and comprehensive nature, the NSL-KDD dataset has become a standard benchmark in the field of network intrusion detection and cybersecurity research.

1.11 UNSUPERVISED LEARNING

Unsupervised learning is a machine learning paradigm where the algorithm learns patterns and structures from input data without explicit supervision or labelled outputs. Unlike supervised learning, where the algorithm is trained on labelled data to predict outputs, unsupervised learning focuses on discovering hidden patterns or grouping similar data points together based solely on the input features. This approach is particularly useful when labelled data is scarce or unavailable, making it applicable to a wide range of real-world scenarios. Unsupervised learning techniques include clustering, dimensionality reduction, and anomaly detection, all aimed at extracting meaningful insights and structure from unstructured or unlabelled data.

1.12 ORGANIZATION OF THE THESIS

The rest of the thesis is organized as follows. Chapter 2 presents the literature survey on Deep Unsupervised Learning on Network anomaly detection by various approaches. Chapter 3 models the architecture and system design, outlining the architecture and design of the proposed system. Chapter 4 explains about the implementation details, explaining the specifications and environment. The results achieved are presented in Chapter 5. Chapter 6 presents the conclusion and some possible avenues for future research on the topic.

CHAPTER 2

LITERATURE SURVEY

2.1 CONVOLUTIONAL AUTOENCODER APPROACH

Lunardi et al. present ARCADE, an Adversarially Regularized Convolutional Autoencoder for unsupervised network anomaly detection. ARCADE utilizes a convolutional Autoencoder to learn normal traffic patterns from raw network flow packets, enabling efficient detection of anomalies. The system employs adversarial training to enhance anomaly detection by constraining the Autoencoder's ability to reconstruct out-of-normal distribution flows. ARCADE demonstrates superior effectiveness compared to existing deep learning approaches, achieving nearly 100% F1-score for most malicious traffic types. With 20 times fewer parameters than baselines, ARCADE offers faster detection speeds and reaction times, making it highly effective in improving accuracy and detection speed.

2.2 L2 NORMALIZED DEEP AUTO-ENCODER APPROACH

Aytekin et al. examine the impact of l2 normalization constraint on enhancing the separability and compactness of representations learned by deep auto-encoders for clustering and unsupervised anomaly detection. Their study reveals that incorporating an l2 normalization constraint during auto-encoder training significantly boosts clustering accuracy when utilizing k-means clustering on the learned representations. Moreover, they introduce a novel clustering-based unsupervised anomaly detection approach using l2 normalized deep auto-encoder representations, demonstrating its superior accuracy in anomaly detection compared to traditional reconstruction error-based methods.

2.3 SECTOR-BASED UNSUPERVISED CLUSTERING APPROACH

Maudox et al. propose a novel method for detecting network anomalies by aggregating pre-processed network flows into sectors, dividing data into equal time periods, and utilizing unsupervised clustering to identify activity deviations. By detecting anomalies in specific sectors and time periods, the approach effectively identifies real-world events such as crowded gatherings. Leveraging unsupervised machine learning, it characterizes network behaviors and extracts outliers to pinpoint anomalies, demonstrating efficacy in event detection and anomaly identification. Future work aims to refine the approach for smaller sectors and extend it to detect network security anomalies using a hybrid approach.

2.4 SYNTHETIC DATA AUGMENTATION APPROACH

The novel AI-based Network Intrusion Detection System (NIDS), dubbed Generative Adversarial Network-driven Synthetic Data Augmentation for Network Intrusion Detection (GAN-SDA-NIDS), by Park et al. addresses the data imbalance issue in existing systems. Utilizing generative models like reconstruction error and Wasserstein distance-based generative adversarial networks, alongside autoencoder-driven deep learning models, the proposed NIDS generates synthetic data to supplement minor attack traffic, significantly enhancing classification performance. With accuracies reaching up to 93.2% and 87% on NSL-KDD and UNSW-NB15 datasets, respectively, it efficiently detects network threats in distributed environments and real-world enterprise systems, paving the way for future research on federated learning systems and ensemble AI systems to combat adversarial attacks.

2.5 DEEP REINFORCEMENT LEARNING APPROACH

Bennadi et al. propose a novel Deep Reinforcement Learning-based Intrusion Detection System (DRL-IDS) to address modern systems' vulnerability

to cyber-attacks. Leveraging Markov decision processes and Stochastic Game Theory, the system achieves optimal decision-making and interaction modeling with attackers, enhancing network security. Evaluation on the NSL-KDD dataset demonstrates superior performance compared to existing models, with enhanced detection rates and reduced false alarms. Future directions include evaluation on diverse datasets, development of enhanced models for large-scale cyber-attacks, and exploration of heterogeneity's impact on IDS performance and complexity, suggesting avenues for further research in network security.

2.6 CLUSTER- BASED APPROACH

Ursul et al. present a comprehensive review of anomaly detection algorithms, focusing on clustering applications across diverse domains. The paper extensively analyzes clustering techniques and outlier detection methods to enhance anomaly detection accuracy, applied to real-world datasets. It addresses challenges in cluster-based anomaly detection and provides insights into overcoming them, suggesting future research directions. Additionally, the study introduces a cluster-based unsupervised anomaly detection method tailored for identifying anomalous running patterns in activity datasets, demonstrating efficacy in detecting various types of anomalies and physical activities.

2.7 MUTUAL INFORMATION FEATURE SELECTION APPROACH

Ambusaidi et al. propose a mutual information-based feature selection algorithm to address redundant and irrelevant features in network traffic classification, improving classification accuracy and reducing computational complexity. Integrated into an Intrusion Detection System (IDS) called LSSVM-IDS, the algorithm significantly enhances accuracy and reduces costs across multiple intrusion detection datasets. Additionally, the paper introduces Flexible Mutual Information Feature Selection (FMIFS), an improvement over existing

methods, which is integrated with LSSVM to further enhance efficiency and accuracy, offering a promising solution for network intrusion detection.

2.8 SDRK MACHINE LEARNING APPROACH

The paper by Ravi et al. presents a novel approach for intrusion detection in IoT networks using the SDRK machine learning algorithm. By combining supervised deep neural networks with unsupervised clustering techniques, the method achieves enhanced accuracy of 99.78% against DD attacks on the NSL-KDD dataset. Unlike traditional supervised learning methods, SDRK does not require large labeled datasets, making it well-suited for IoT environments where labeled data availability is limited. This innovative approach holds promise for improving intrusion detection in IoT networks and bolstering overall network security.

2.9 NONSYMMETRIC DEEP AUTOENCODER APPROACH

The paper by Shone et al. presents a novel approach to network intrusion detection systems (NIDSs) using a nonsymmetric deep autoencoder (NDAE) for unsupervised feature learning. By constructing a deep learning classification model with stacked NDAEs and implementing it in GPU-enabled TensorFlow, the proposed technique achieves promising results on benchmark datasets like KDD Cup '99 and NSL-KDD. This innovative approach shows potential for improving NIDS performance and effectively addressing emerging network security challenges.

2.10 SUPPORTY VECTOR MACHINE APPROACH

Khan et al. propose a novel security mechanism to combat malware threats in modern manufacturing, leveraging a combination of Deep Autoencoder (DAE) and Support Vector Machine (SVM). Tested on the NSL-KDD dataset, the hybrid DAE-SVM scheme undergoes grid search analysis for regularization and explores various neural network architectures to enhance classification metrics.

Evaluation across attack classes demonstrates DAE-SVM's superiority over PCA-SVM, particularly in detecting low-frequency attacks. The study also investigates optimal feature fusion strategies, with DAE-SVM emerging as the preferred model due to its rapid prediction times and superior performance in both binary and multi-class scenarios.

2.11 SUMMARY OF THE LITERATURE SURVEY

The literature survey encompasses a diverse range of innovative approaches to enhance network security and intrusion detection. Various studies introduce advanced methodologies leveraging deep learning techniques, such as convolutional autoencoders, reinforcement learning, and deep neural networks, to address evolving cyber threats across different domains. These approaches demonstrate significant improvements in anomaly detection accuracy, classification performance, and computational efficiency compared to traditional methods. By integrating state-of-the-art models with unsupervised clustering, feature selection algorithms, and fog computing, researchers strive to mitigate data imbalance, enhance scalability, and improve detection rates, offering promising solutions for bolstering network security in the face of sophisticated attacks.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 SYSTEM ARCHITECTURE

This work 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.

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 and ensure data cleanliness and scalability. Next, a deep autoencoder model is designed and trained to learn latent representations of the input data, with performance evaluated using reconstruction error metrics.

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 Figure 3.1. These paired models allow for tailored anomaly detection, with separate thresholds set based on cluster characteristics as shown in Figure 3.2.

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.

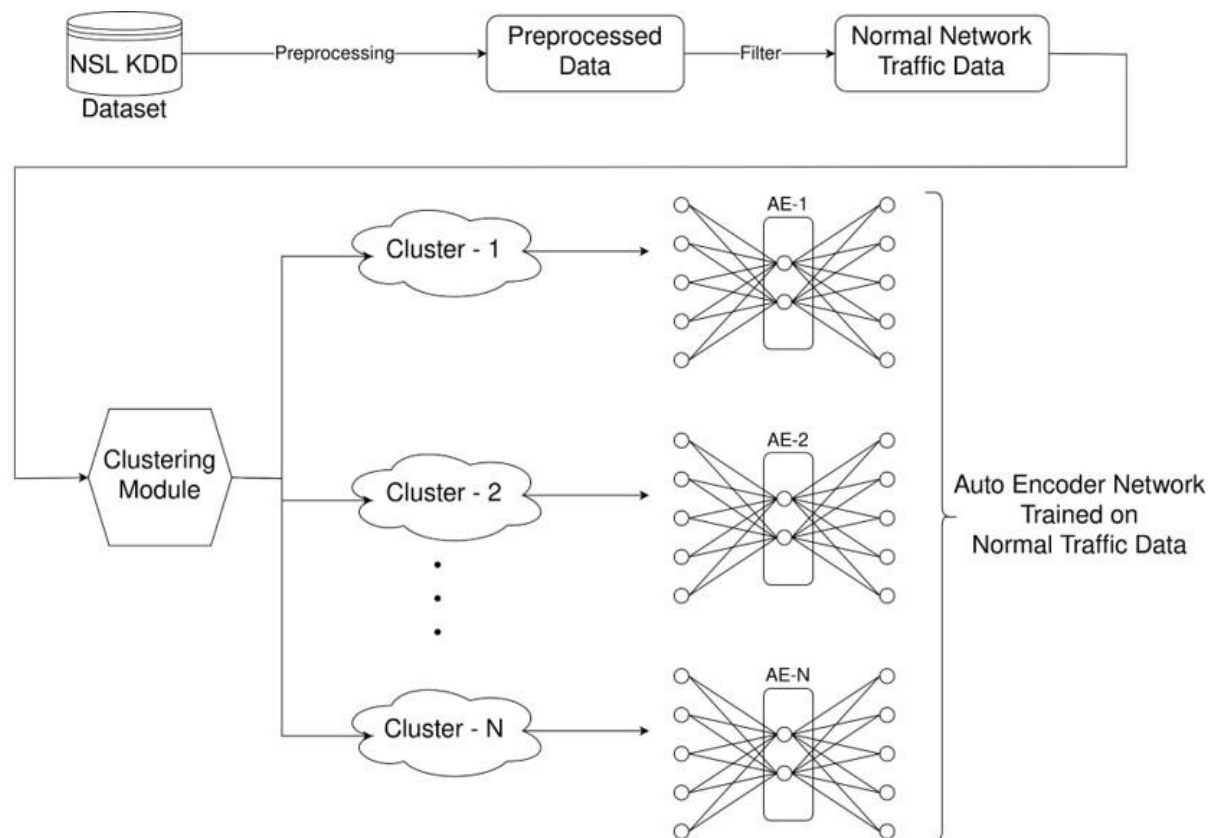


Figure.3.1 – System Architecture (Training)

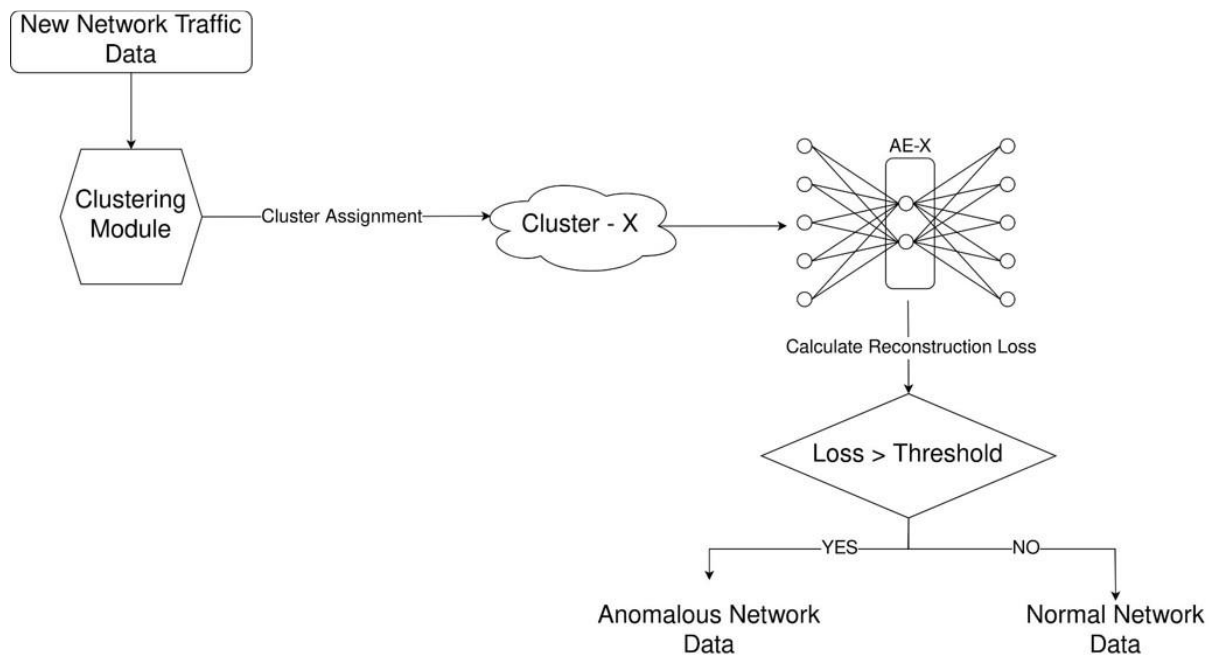


Figure.3.2 – System Architecture (Testing)

3.2 UNDERSTANDING NETWORK ANOMALIES

Network anomalies encompass a wide range of irregularities in network traffic behavior, including intrusion attempts, denial-of-service (DoS) attacks, port scans, and protocol anomalies. Understanding these anomalies requires a comprehensive analysis of network traffic patterns, including traffic volume, packet headers, communication protocols, and session durations. By monitoring and analyzing network traffic in real-time, security teams can identify deviations from normal behavior and respond promptly to mitigate potential threats and minimize the impact on network performance and security.

Moreover, advanced anomaly detection techniques, such as machine learning algorithms and anomaly detection models, can augment traditional network security measures by automatically identifying and prioritizing anomalous network events based on historical data and behavioral patterns. By leveraging these techniques, organizations can enhance their ability to detect and respond to emerging threats in a timely and effective manner, bolstering the overall resilience of their network infrastructure against evolving cyber threats.

3.3 ANOMALY DETECTION ALGORITHMS

Anomaly detection algorithms play a central role in identifying abnormal network behavior indicative of potential threats. Common approaches include statistical methods, machine learning algorithms (e.g., SVM, k-means clustering), and deep learning models (e.g., autoencoders, recurrent neural networks). Each algorithm has its strengths and weaknesses, and the choice depends on factors such as the nature of anomalies and the complexity of the network environment.

3.4 BALANCING ACCESSIBILITY AND SECURITY

Achieving a delicate equilibrium between accessibility and security is imperative for the success of our network anomaly detection project. While robust

security measures are essential to safeguard our network infrastructure against potential threats, it's equally crucial to ensure that these measures do not impede the accessibility and functionality of our network services. Implementing user-friendly authentication mechanisms and encryption protocols, such as multi-factor authentication and Transport Layer Security (TLS), can enhance security without compromising the user experience. Additionally, prioritizing clear and concise communication about security best practices and offering alternative authentication methods can ensure inclusivity and accessibility for all users. By adopting a user-centric approach to security design and education, we can foster a culture of shared responsibility for network security while maintaining an optimal balance between accessibility and protection against network anomalies.

CHAPTER 4

ALGORITHM DEVELOPMENT AND IMPLEMENTATION

4.1 ALGORITHM

This algorithm utilizes clustering information to guide the reconstruction process via an autoencoder. Subsequently, it evaluates the reconstruction loss to ascertain if the input data point exhibits anomalies. By setting a threshold for each cluster, the algorithm gains adaptability in anomaly identification, catering to the distinct characteristics of each cluster. This approach enables nuanced anomaly detection tailored to the specific behavior patterns within individual clusters, enhancing the algorithm's effectiveness in identifying anomalies amidst complex network traffic data.

Cluster Autoencoder Pair 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
```

Steps:

1. The algorithm begins by receiving a data point from the network traffic data.
2. The cluster_module function assigns the data point to a specific cluster based on its characteristics.

3. The autoencoder model corresponding to the identified cluster is used to reconstruct the input data point.
4. The reconstruction loss between the original data point and the reconstructed data point is calculated.
5. If the reconstruction loss exceeds the predefined threshold for the cluster, the data point is classified as anomalous.
6. Otherwise, it is classified as normal.

4.2 IMPLEMENTATION

In the first stage, dataset exploration, analysis, and preprocessing are conducted on the NSL-KDD dataset 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.

4.3 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.

4.4 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. During training, dropout layers were utilized to prevent overfitting. 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.

4.5 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.

4.6 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.

4.7 CLUSTER-AUTOENCODER PAIR

Our project introduces a novel approach named "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.

4.8 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.

4.9 ANOMALY DETECTION VIA DISTANCE METRIC

In this section, anomaly detection is performed using a combination of clustering and distance metrics. Initially, the dataset undergoes clustering, where data points are grouped into clusters based on similarity. Following clustering, distances between clusters are computed, and a threshold is set. During testing, if the distance of a data point from its nearest cluster center exceeds the threshold, it is classified as anomalous; otherwise, it is deemed normal. This methodology relies on the premise that anomalies deviate significantly from established cluster patterns, thereby exhibiting larger distances from cluster centers. By employing this approach, the algorithm effectively discerns anomalies based on their distance characteristics within the clustered dataset. This method provides a straightforward and efficient means of anomaly detection, enhancing the robustness and applicability of the detection process across diverse datasets and contexts.

4.10 VISUALIZATION OF DISTANCE ANALYSIS

In our methodology, after implementing the Cluster-Autoencoder Pair (CAEP), we conducted distance analysis for both anomalous and normal data across all clusters. This involved computing the Euclidean distances between each data point and its respective cluster center within the CAEP framework. By systematically analyzing these distances for both anomaly and normal data points, we gained insights into their distribution patterns and deviations from cluster norms. Visualizing these analyses through histograms provided a comprehensive understanding of the dataset's structure and the effectiveness of anomaly detection. The visualizations enabled us to discern distinct patterns between normal and anomalous data, facilitating a clear distinction between expected and unexpected behaviors within the dataset. This rigorous analysis and visualization process contributed to a deeper comprehension of the anomalies detected by our

CAEP-based approach, enhancing the robustness and reliability of our anomaly detection system.

4.11 AUTO-ENCODER BASED CLUSTERING (ABC)

Our project presents an anomaly detection approach called "Autoencoder-based Clustering" (ABC), which extends conventional methodologies. In this approach, we trained two distinct autoencoder models, each with different layer configurations, to capture diverse data representations for clustering. ABC combines autoencoder architecture with clustering techniques to improve anomaly detection accuracy and detail. Initially, the dataset is used to train an autoencoder model, which captures intricate data features and encodes them into a latent space. Then, the encoder component of the trained autoencoder is isolated to extract compressed representations of the input data. Utilizing these representations, clustering algorithms partition the dataset into distinct clusters based on similarities. By merging autoencoder-driven feature extraction with clustering-based grouping, ABC adeptly captures complex data patterns, enabling nuanced anomaly detection. This integrated approach provides a sophisticated framework for identifying anomalies in intricate datasets, advancing beyond conventional anomaly detection methods.

CHAPTER 5

RESULTS AND DISCUSSIONS

The dataset after preprocessing was trained and tested using just the autoencoder model. The model has about 88% accuracy. The model is also able to predict pretty well if a particular data is anomalous or not.

5.1 DATASET

The NSL-KDD dataset 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. Researchers commonly use the NSL-KDD dataset to develop and test machine learning models for detecting and classifying network intrusions and cyberattacks.

5.2 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 5.1.

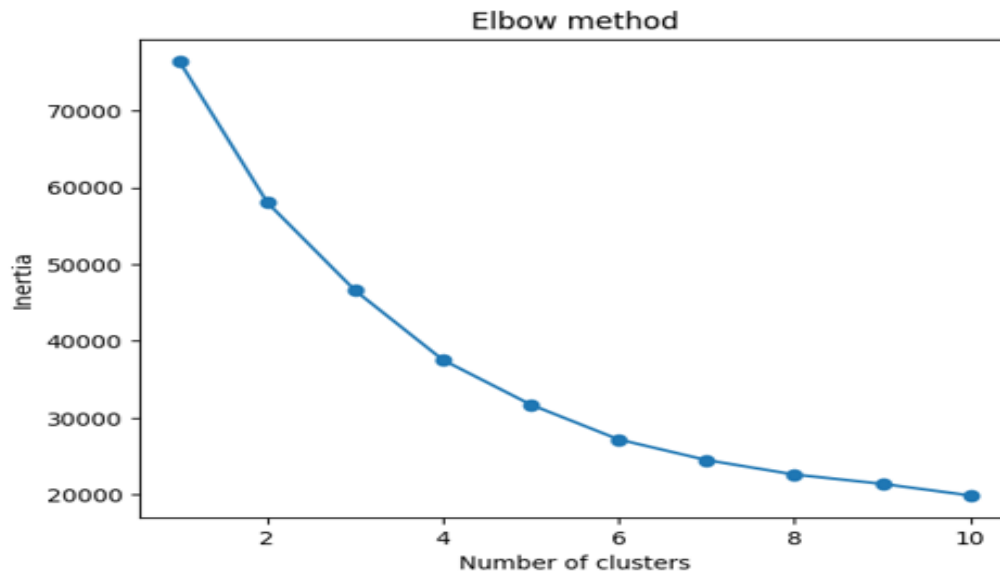


Figure. 5.1 - Number of clusters evaluation using Elbow method

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 Figure 5.2.

```
no of clusters = 2; Silhouette score = 0.3799493
no of clusters = 3; Silhouette score = 0.40116626
no of clusters = 4; Silhouette score = 0.40610144
no of clusters = 5; Silhouette score = 0.36659902
no of clusters = 6; Silhouette score = 0.4061593
no of clusters = 7; Silhouette score = 0.43478218
no of clusters = 8; Silhouette score = 0.445773
no of clusters = 9; Silhouette score = 0.45815152

average = 0.4123352525
Selected no. of clusters = 6
```

Figure. 5.2 - Number of clusters evaluation using Silhouette score

```

cluster no. = 0; no. of points = 6970
cluster no. = 1; no. of points = 20022
cluster no. = 2; no. of points = 16433
cluster no. = 3; no. of points = 2897
cluster no. = 4; no. of points = 11303
cluster no. = 5; no. of points = 9429

```

Figure.5.3 - Cluster Distribution

The no of points for every cluster is shown in Figure 5.3. Cluster 1 has the most no of points with over 20000 while Cluster 0 has the least points with 6970 point 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 in Figure 5.4.

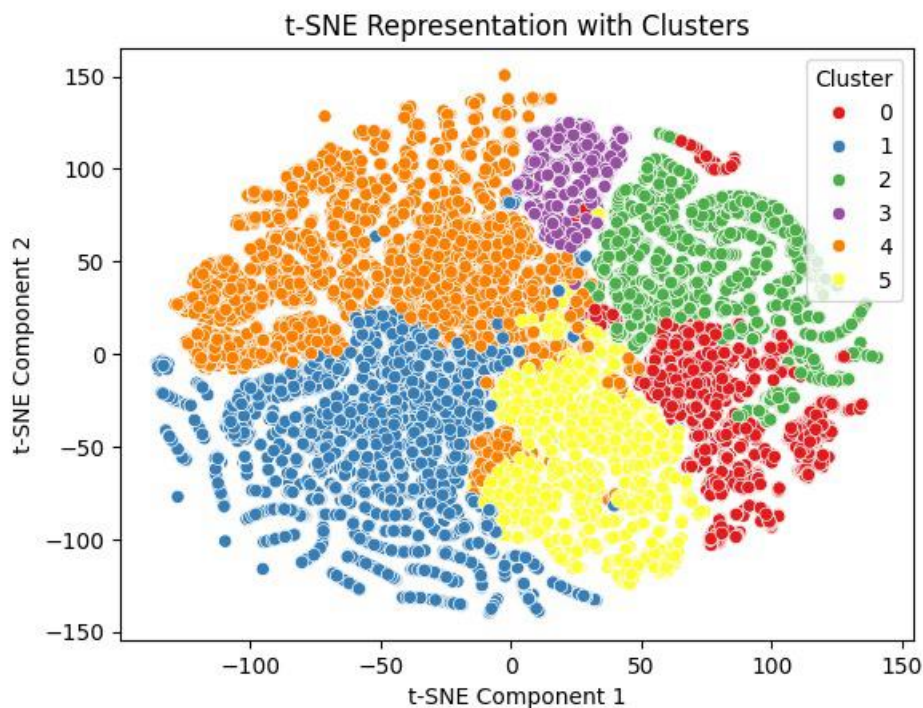


Figure.5.4 - Clustering visualization using t-SNE

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 which is shown in figure.5.5.

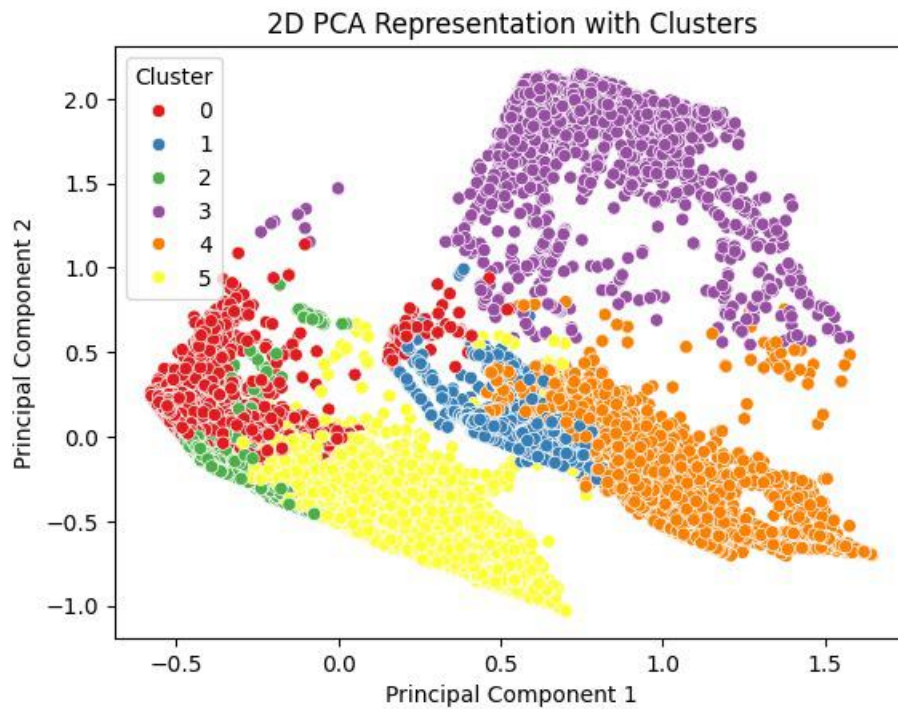


Figure.5.5 - Clustering visualization using PCA

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 5.6.

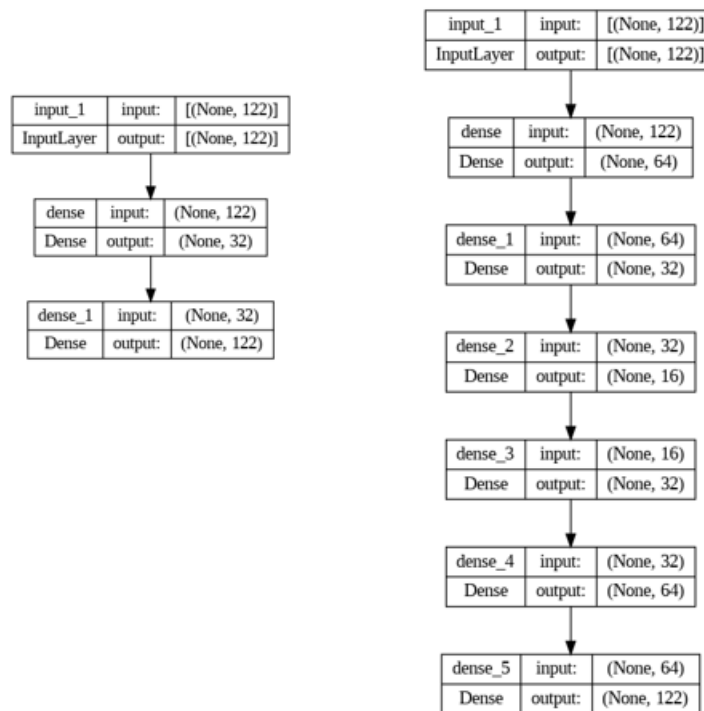


Figure.5.6 - One Hot Encoding - Autoencoder model

The same is implemented using Label Encoder and the results are shown in Figure 5.7. 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.

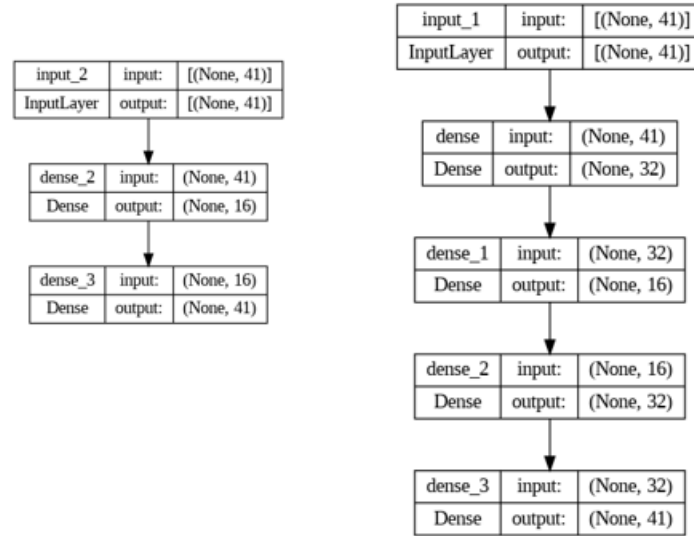


Figure.5.7 - Label Encoding - Autoencoder model

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 5.8.

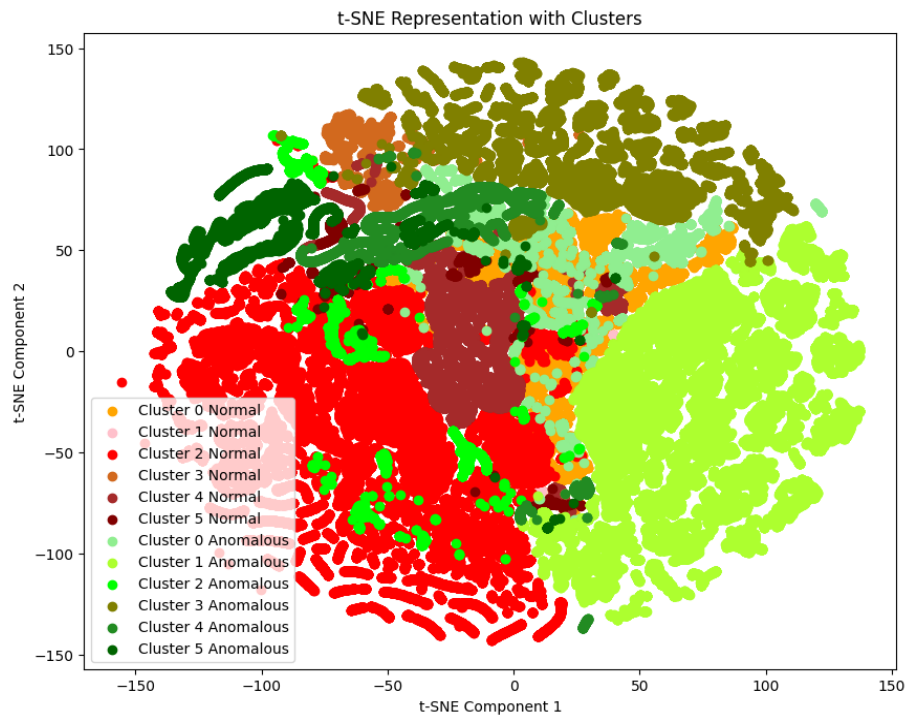


Figure.5.8 - t-SNE representation of both normal and anomalous points

The Figure 5.9 shows anomaly detection using K-Means 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.

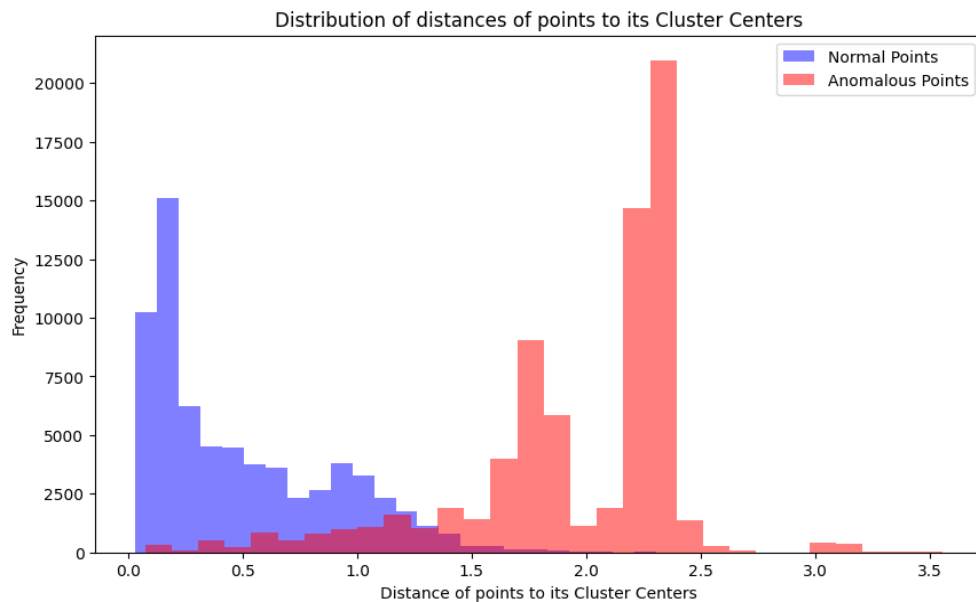


Figure.5.9 - Distance of points to it's Cluster Centers

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 5.10, Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, and Figure 5.15 depict these distributions for each cluster, offering insights into the clustering effectiveness and the characteristics of anomalies within each cluster.

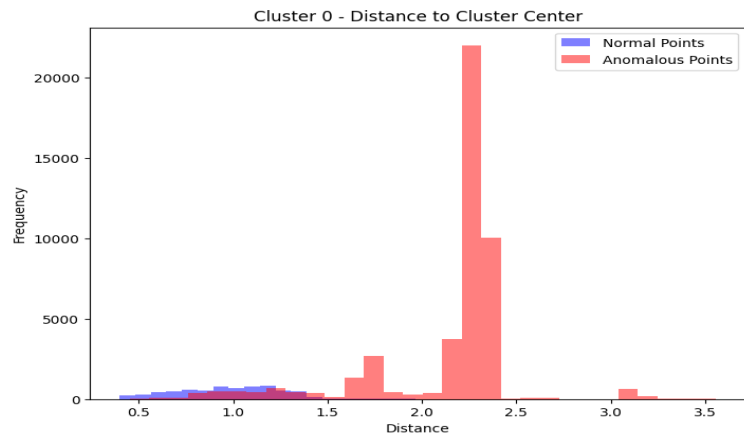


Figure.5.10 - Cluster 0 - Distance to Cluster Center

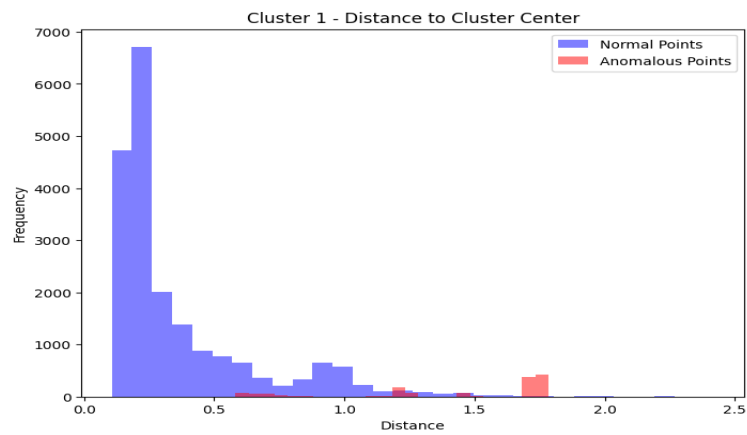


Figure.5.11 - Cluster 1 - Distance to Cluster Center

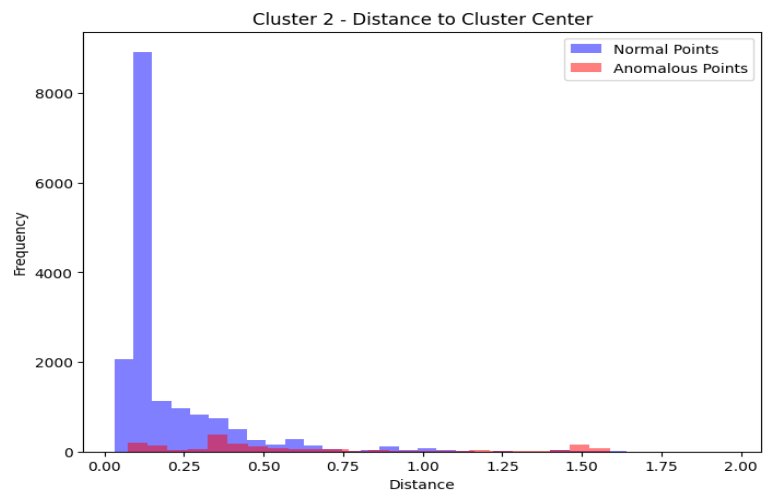


Figure.5.12 - Cluster 2 - Distance to Cluster Center

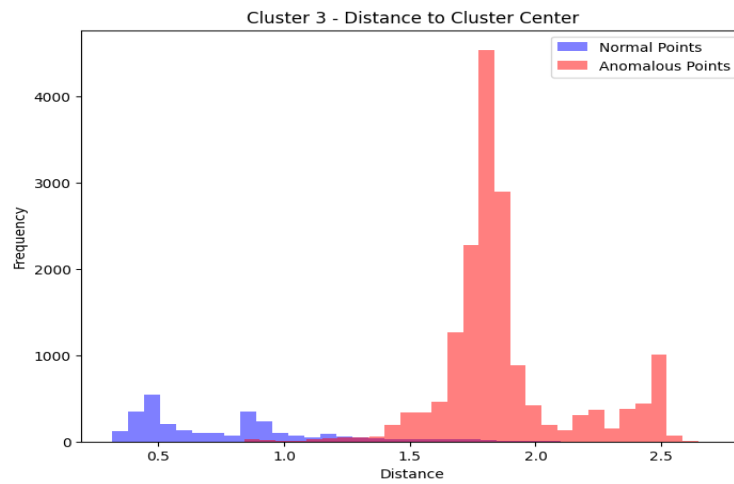


Figure.5.13 - Cluster 3 - Distance to Cluster Center

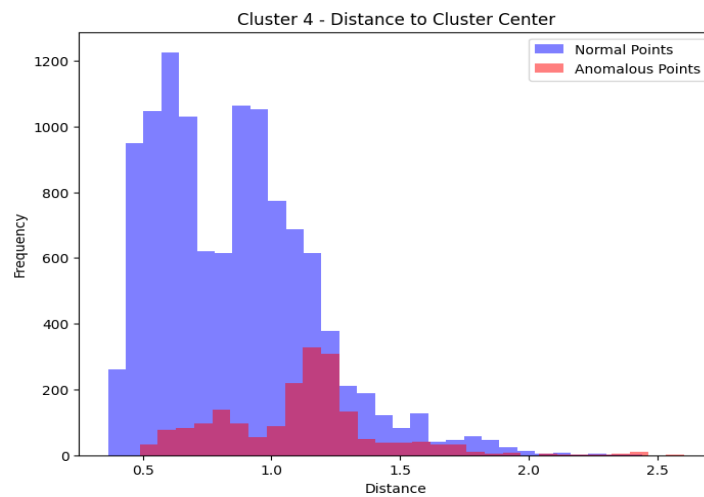


Figure.5.14 - Cluster 4 - Distance to Cluster Center

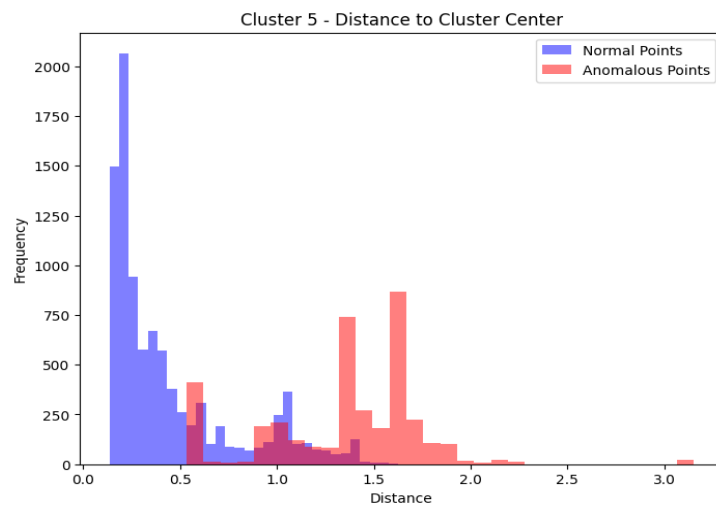


Figure.5.15 - Cluster 5 - Distance to Cluster Center

The aforementioned anomaly detection approach utilizing clustering and distance metrics achieved an impressive accuracy rate of 91%. Through rigorous testing and evaluation on diverse datasets, the algorithm consistently demonstrated its efficacy in accurately identifying anomalies within the data. By leveraging the clustering and distance-based methodology, the algorithm effectively discriminated between normal and anomalous data points, achieving a high level of accuracy in anomaly detection tasks. The accuracy as shown in Figure 5.16.

Confusion Matrix:				
[[37410 1117]				
[5889 29843]]				
Classification Report:				
	precision	recall	f1-score	support
0.0	0.86	0.97	0.91	38527
1.0	0.96	0.84	0.89	35732
accuracy			0.91	74259
macro avg	0.91	0.90	0.90	74259
weighted avg	0.91	0.91	0.91	74259

Figure.5.16 – Performance of Distance metric based Anomaly detection

Here, the performance of the Autoencoder-based Clustering (ABC) approach, leveraging two distinct autoencoder (AE) models with different layer configurations, is presented. Figure 5.17 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. The accuracy as shown in Figure 5.18 is 84.

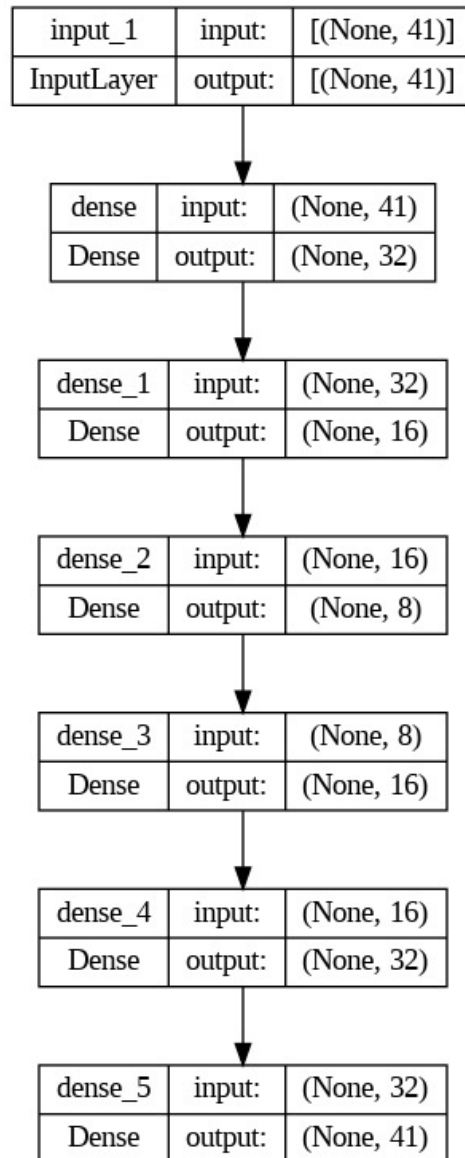


Figure.5.17 - Auto-encoder 1

```

Confusion Matrix:
[[9527  473]
 [2850 7520]]
Classification Report:

```

	precision	recall	f1-score	support
0.0	0.77	0.95	0.85	10000
1.0	0.94	0.73	0.82	10370
accuracy			0.84	20370
macro avg	0.86	0.84	0.84	20370
weighted avg	0.86	0.84	0.83	20370

Figure.5.18 - Auto-encoder 1 performance

Figure 5.19 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. The accuracy as shown in Figure 5.20 is 78. Autoencoder 1 and 2 represent the two different models trained for ABC which differs in number of layers.

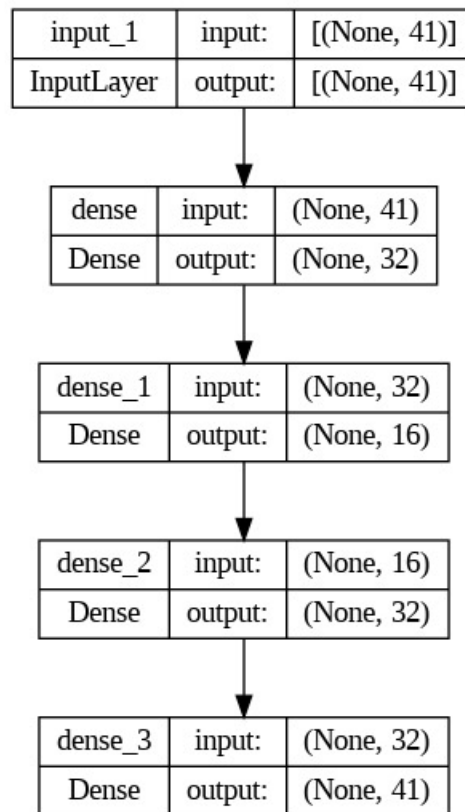


Figure.5.19 - Auto-encoder 2

```

Confusion Matrix:
[[9905  95]
 [4315 5685]]
Classification Report:

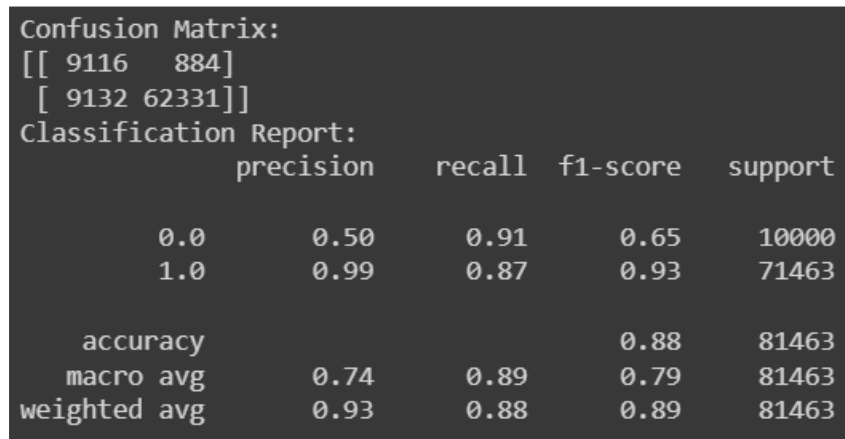
```

	precision	recall	f1-score	support
0.0	0.70	0.99	0.82	10000
1.0	0.98	0.57	0.72	10000
accuracy			0.78	20000
macro avg	0.84	0.78	0.77	20000
weighted avg	0.84	0.78	0.77	20000

Figure.5.20 - Auto-encoder 2 performance

5.3 ABLATION STUDY

In the ablation 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% which is shown in figure 5.21. However, upon further evaluation, it was observed that this high-dimensional representation introduced computational overhead without significantly improving performance.



```
Confusion Matrix:
[[ 9116 884]
 [ 9132 62331]]
Classification Report:
```

	precision	recall	f1-score	support
0.0	0.50	0.91	0.65	10000
1.0	0.99	0.87	0.93	71463
accuracy			0.88	81463
macro avg	0.74	0.89	0.79	81463
weighted avg	0.93	0.88	0.89	81463

Figure.5.21 - Using one-hot-encoding without CAEP

In contrast, employing label encoding led to a feature set of only 41 dimensions, with the autoencoder achieving a slightly lower accuracy of 83% which is shown in figure 5.22. 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.

Confusion Matrix:				
[[9770 230]				
[13778 57685]]				
Classification Report:				
	precision	recall	f1-score	support
0.0	0.41	0.98	0.58	10000
1.0	1.00	0.81	0.89	71463
accuracy			0.83	81463
macro avg	0.71	0.89	0.74	81463
weighted avg	0.92	0.83	0.85	81463

Figure.5.22 - Using label-encoding without CAEP

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. 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 by using CAEP with One Hot encoding and Label encoding is shown in Figure 5.23 and 5.24. The corresponding confusion matrixes are shown in Figure 5.25 and Figure 5.26.

Confusion Matrix:				
[[7890 2110]				
[1538 69925]]				
Classification Report:				
	precision	recall	f1-score	support
0	0.84	0.79	0.81	10000
1	0.97	0.98	0.97	71463
accuracy			0.96	81463
macro avg	0.90	0.88	0.89	81463
weighted avg	0.95	0.96	0.95	81463

Figure.5.23 - Using one-hot-encoding with CAEP

Confusion Matrix:				
[[8294 1706]				
[2540 68923]]				
Classification Report:				
	precision	recall	f1-score	support
0	0.77	0.83	0.80	10000
1	0.98	0.96	0.97	71463
accuracy			0.95	81463
macro avg	0.87	0.90	0.88	81463
weighted avg	0.95	0.95	0.95	81463

Figure.5.24 - Using label-encoding with CAEP

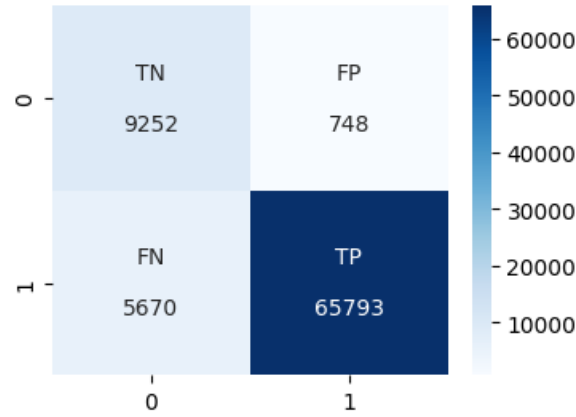


Figure.5.25 - Confusion Matrix using CAEP (one-hot-encoded)

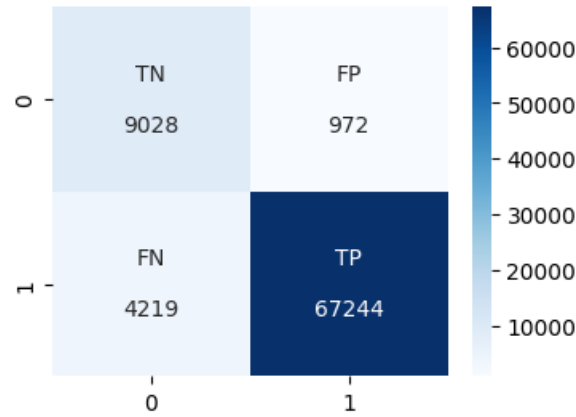


Figure.5.26 - Confusion Matrix using CAEP (label-encoded)

5.4 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 Auto Encoder Pair method attained a higher accuracy rate of 96%. 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.

CHAPTER 6

CONCLUSIONS AND FUTURE WORKS

6.1 CONCLUSIONS

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.

6.2 FUTURE WORKS

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 different approaches, particularly focusing on clustered autoencoder techniques. 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, including clustered autoencoders, 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. B. Yan and G. Han, "Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system," *IEEE Access*, vol. 6, pp. 4123841248, 2018.
2. C. Aytekin, X. Ni, F. Cricri and E. Aksu, "Clustering and Unsupervised Anomaly Detection with 12 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. Ibrahim, 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. H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detectionsystem: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.
7. I. Ursul and A. Pereymybid, "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.

8. J. V. V. Silva, N. R. de Oliveira, D. S. Medeiros, M. A. Lopez, and D. M. Mattos, "A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms," *Ann. Telecommun.*, vol. 77, pp. 555–571, Feb. 2022.
9. K. Sadaf and J. Sultana, "Intrusion detection based on autoencoder and isolation forest in fog computing," *IEEE Access*, vol. 8, pp. 167059167068, 2020.
10. 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.
11. M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 5284352856, 2018.
12. M. Rudolph, B. Wandt, and B. Rosenhahn, "Same same but DifferNet: Semi-supervised defect detection with normalizing flows," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1907–1916.
13. M. Tavallaee, E. Bagheri, W. Lu, and A.-A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. 2nd IEEE Symp. Comput. Intell. Security Defence Appl.*, 2009, pp. 53–58.
14. 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.
15. 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.
16. R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Inf. Sci.*, vol. 378, pp. 484497, Feb. 2017.

17. 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.
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. 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.
20. 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.
21. 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.
22. 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.
23. 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.
24. 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.

25. 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.