National College of Ireland

Securing Cloud Environments Through Real-Time Network Monitoring System for Detecting Network Attacks using Advanced Deep Learning Methods

MSc Research Project
MSc in Cloud Computing

Venkateshwarlu Vanga
Student ID: x22158952

School of Computing
National College of Ireland

Supervisor: Shaguna Gupta

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Venkateshwarlu Vanga
Student ID:	x22158952
Programme:	MSc in Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Shaguna Gupta
Submission Due Date:	14/12/2023
Project Title:	Securing Cloud Environments Through Real-Time Network Monitoring System for Detecting Network Attacks using Advanced Deep Learning Methods
Word Count:	7440
Page Count:	30

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Venkateshwarlu Vanga
Date:	26th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing Cloud Environments Through Real-Time Network Monitoring System for Detecting Network Attacks using Advanced Deep Learning Methods

Venkateshwarlu Vanga
x22158952

Abstract

In the domain of cloud computing, the widespread adoption of shared resources and remote services increases security vulnerabilities, particularly from hostile data packets threatening data confidentiality, integrity, and availability. As cloud computing becomes integral to business operations, organizations must adopt advanced strategies and technologies to mitigate these risks. This study explores deep learning algorithms for anomaly detection in cloud-based systems, where anomalies signal potential threats to data security and system integrity. Focusing on the effectiveness of Graph Neural Network (GNN), Autoencoder, and Recurrent Neural Network (RNN), the research adopts a comprehensive methodology, beginning with anomaly data collection from reliable sources. Preprocessing steps ensure data quality and balance, while feature engineering techniques like label encoding and principal component analysis (PCA) optimize data representation and reduce dimensionality for enhanced efficiency. Among the evaluated algorithms, the Autoencoder emerges as most effective in anomaly detection within cloud environments, achieving 99.99% accuracy. Its superior sensitivity and specificity effectively minimize false positives to 0.001, accurately identifying anomalies and contributing significantly to automated anomaly detection software development. This research advances the field of anomaly detection in cloud computing, providing insights into the relative merits of various deep learning approaches and their practical applications in ensuring cloud data security.

1 Introduction

Network intrusion is a common cyber problem that has been identified to increase in recent times, particularly in cloud computing environments. As the literature indicates, cyber-attacks have evolved tremendously, with hackers employing sophisticated approaches based on advances in hardware, software, or network topologies. This evolution has contributed to the continuous prevalence of malicious network intrusion, especially in cloud systems where shared resources and services are a norm. Understanding this, some studies have informed that the rapid progression in modern technological infrastructure, such as the Internet and increased application of cloud computing, has extensively diversified the global network system. With the extension of the network system at a global level, including cloud infrastructures, faults and attacks have been increasing frequently,

impacting users' experience and burdening them with economic and reputational damages.

Therefore, experts have been continuously exploring options to detect these anomalies, particularly in cloud-based systems. However, as a first-line of defense, users have been using firewalls to detect the anomaly and ensure that their network system has been working correctly. Based on the previous information, it has been acknowledged that anomalies in the network system due to tremendous cyber-attacks have become a common issue for users. In this regard, as a primary protective component, firewalls are installed for safe and reliable network function, including in cloud environments. However, as studies implied, with the rapid enhancement of sophistication in cyber-attacks, it has become challenging for users to rely only on firewalls, especially in cloud systems where the architecture and data flow are more complex.

Therefore, a relative approach to the second-line of defense has become a persistent approach. As Lin et al. (2019) explained, the "intrusion detection system" (IDS) has gained immense popularity among cloud service providers and users as the secondary defense component, which has further improved the network security system. The consideration of the IDS system as an advanced security measure for users' network systems, including cloud-based networks, has become a valuable approach that monitors the network traffic and detects anomalies in real-time. According to the information further provided by Lin et al. (2019), the detection of anomalies in the network traffic implies abnormalities that adversely impact the network system and subsequently deviate the pattern from ordinary traffic. A suitable understanding of these anomalies can enable the user to successfully detect network issues without any damage, especially in cloud-based applications.

1.1 Motivation

Network anomaly detection is a complicated process and has become more advanced with sophistication in the technique typically used by hackers to attack users. Considering this fact, studies have revealed that there have been successive approaches ensured by organizations to improve this "intrusion detection system" (IDS) or "intrusion prevention system" (IPS) to ensure that computer networks have been functioning correctly. More importantly, in recent times, with technological advances in organizations, enterprise-based users, particularly those utilizing cloud computing, are facing constant dilemmas in securing their network systems from cyber-attacks. Therefore, recent advances by industrial experts and researchers have explored the paradigm of machine learning and deep learning algorithms, which are used as advances to build the IDS system. The intrusion detection system (IDS) that has been built using machine learning (ML) or deep learning (DL) algorithms has shown increased accuracy and effectiveness in the detection process, particularly within cloud computing environments.

Depending on the indication of information regarding the effectiveness of algorithmic methods in enhancing intrusion detection systems, a significant pattern in the detection process is identified. As per the information provided by Kasongo and Sun (2020), the built-in approach to the IDS system using ML or DL methods has shown a potential decrease in performance within the high-dimensional data space. Therefore, the recent advances in anomaly detection in computer networks, including cloud systems, based on the

advanced intrusion detection system have focused on the feature selection process using integrated datasets that are prone to provide exclusive features to promote more excellent classification and detection accuracy. In this regard, one of the potential approaches to efficient feature selection from datasets implies the use of the "UNSW-NB-15" intrusion prediction dataset. The contribution of this dataset has marked a significant approach in the detection process and reduces the issues aligned with highly "imbalanced" datasets.

Understandably, network intrusion detection has become a significant consideration in contemporary research, where experts have been investigating a modified approach that can suitably improve detection efficiency, especially in cloud-based systems. Depending on the information presented above, the network-based intrusion detection system has been modified to an enhanced level to ensure safe and reliable network surfing and cloud computing applications in organizations and among individual users. The safe prediction of the intrusion based on the features extracted from datasets is a common approach while integrating ML and DL methods into the prediction and classification process.

However, as discussed earlier, the imbalance present in existing datasets has shown a decrease in effectiveness as well as the efficiency level. More inclined to the fact that standard algorithms, when utilizing these datasets, provide biased results, thus reducing the reliability of these models in the detection process. Therefore, it is always imperative to use a reliable and well-balanced dataset for the detection of network traffic, particularly in cloud computing environments. Moreover, while using imbalanced datasets, it has been observed that conventional models have shown necessary limitations in the detection process, particularly machine learning models. Therefore, the recent focus has been given to deep learning models that provide highly accurate prediction or detection results compared to traditional methods and models. Therefore, this study has become divergent toward the exploration of network intrusion detection systems using deep learning models while extracting features from the UNSW-NB-15 datasets.

1.2 Research Objectives

In this research, we aim to enhance cloud security through the following key objectives:

- To develop and refine Deep Learning models specifically tailored for anomaly detection in cloud computing environments.
- To conduct a comprehensive comparative analysis of the trained models based on key performance metrics such as accuracy, sensitivity, and false positive rates for predicting the network attacks.
- To Develop a web-based user interface that integrates the most effective Deep Learning model for real-time monitoring and analysis of network packets for securing the cloud systems.

1.3 Research Question

This research focuses on the intersection of Deep learning and cloud security and the research question is formulated as follows:

- Which Deep Learning algorithm accurately identifies the Network attack and how it can be implemented in a web-based interface to improve anomaly detection and real-time security monitoring in cloud computing environments?

This research addresses the challenge of sophisticated cyber threats in cloud infrastructures, where traditional, static security systems fall short. The goal is to develop a dynamic, intelligent system using Deep Learning algorithms like CNNs or RNNs, known for effective pattern recognition and anomaly detection. By integrating these into a user-friendly web interface, the project aims to significantly enhance real-time monitoring and response capabilities in cloud security.

1.4 Structure of Report

In this research report, we begin with an Introduction, setting the context and objectives of our study on enhancing cloud security using deep learning models. This is followed by a Literature Review, where we critically analyze existing work and identify gaps our research aims to fill. The Methodology section outlines our research approach, focusing on the specific deep learning techniques used. In Design Specification, we detail the architectural framework of our solution, leading into the Implementation section, where we describe the practical development of our web-based interface and model integration. The Results section presents the outcomes of our implementation, followed by a Discussion where these results are interpreted and analyzed in the context of our initial objectives. The report concludes with a Conclusion and Future Work section, summarizing our findings and suggesting directions for further research in this domain.

2 Related Work

In the current chapter, the discussion has progressed to understanding the potential of the UNSW-NB-15 dataset to provide exclusive features for the network intrusion detection process. In this regard, different classification models - traditional and advanced models have been explored to acknowledge their efficiency in the detection process using the features from this dataset. The review of the information from previous studies will help in comparing the efficiency of the models in the accurate prediction of network intrusion.

2.1 UNSW-NB-15 Network Intrusion Detection Dataset

In recent advances in anomaly detection in computer networks, experts have prioritized the use of an improved dataset - UNSW-NB-15, which has been introduced by the "Intelligent Security Group" (ISG) UNSW Canberra, Australia (Nour Moustafa Abdelhameed Moustafa, 2021). As per the review of studies, thorough research has been performed in the network "intrusion detention system," especially in anomaly detection, which has become a significant cause of novel attacks compared to signature detection. In this approach, the most common datasets predominantly used were KDD99 or NDLKDD datasets, the result of which cannot be considered satisfactory due to a deficit in featuring the modern footprint attack, absence of regular network traffic common in modern times, and imbalance in the distribution of "training" and "testing" sets Moustafa and Slay (2016). As explained by Moustafa and Slay (2016), these three issues identified with

the existing datasets are contemporarily addressed by the modern UNSW-NB-15 dataset. A typical explanation of the dataset indicates that the set contains 9 distinct types of patterns demonstrating modern attacks and normal traffic and accordingly contains nearly 49 attributes. These attributes typically comprise flow-based between the host and the "network packets inspection" that discriminate among observations as normal and abnormal (anomaly). According to the experimental result from testing the dataset, it has been demonstrated that although the dataset is complex, it can be regarded as a benchmark in evaluating network intrusion in modern times.

It has already been explained that the dataset contains 9 different categories of attack types that have been named "Analysis," "Fuzzers," "exploits," "ShellCode," "Reconnaissance," "DOS," "Backdoors," "ShellCode," and "Worms." Based on these attack types, the training UNSW-NB-15 dataset is used to identify the novel attacks that have become more sophisticated in recent years. As explained in the study by Kanimozhi and Jacob (2019), the selected system in the study has used 4 different features of the dataset, which are extracted from a total of 45 and further classified using ML classifiers such as Random Forest (RF) and Decision Tress (DT). As per the understanding of the result, this dataset provides improved accuracy for existing and novel or zero-day attacks.

2.2 Machine Learning-based Methods for Detecting Network Attacks to Secure Cloud System

Over the years, extensive research has been performed to evaluate the intrusion in computer networks, where advances in both detection methods and datasets for feature extraction are identified. As explained by Kabir et al. (2022), these methods can be identified as a single classical approach, hybrid or ensemble structure that is in practice for the development of the intrusion model. Coming to an understanding of the contribution of these models to network intrusion detection, the suitability of machine learning models is primarily explored Nawir et al. (2018). Based on the study conducted by Kabir et al. (2022), two different machine learning stacking classifiers with Extra Trees (ET) have been used along with the "Mutual Information Gain" feature that can provide better accuracy in the result. In this approach, the UNSW_NV_15 dataset is used, which contains packet features and attributes, thus enhancing the detection of novel attacks. As per the comparison of the accuracy level, it has been identified that the proposed model has achieved 96.24% accuracy compared to existing models, thus specifying the supremacy of both the dataset and stacking models. Another study introduced by Kasongo and Sun (2020) explained that a wide gap in the detection accuracy of existing datasets had reduced the efficiency of many machine-learning-based IDS systems for computer networks. This has been identified with the increased false-positive rate of the trained model with imbalanced datasets. Upon exploring the issue, Kasongo and Sun (2020) have implied an analytical approach by applying UNSW_NV_15 datasets for ML models such as "Support Vector Machine" (SVM), "K-Nearest Neighbour" (KNN), "Logistic Regression" (LR) and "Decision Trees" (DT). Moreover, a filter-based algorithm for feature reduction, which indicates the XGBoost ML algorithm, has been used. According to the understanding of the experimental outcome, the "filter-based feature reduction" algorithm has enabled the trained model, such as decision trees, to increase the accuracy level by 90.85% in the case of "binary classification" schemes.

The growth of network attacks at an exponential rate over the years has become a significant concern among network service providers, users, and experts Sonule et al. (2020). As informed by Shushlevska et al. (2022), the increase in network surveillance using the intrusion detection system, therefore, has become a mandatory approach, which has also been identified to be improved in recent years both in models responding conditions and dataset infrastructure that can potentially reduce false alarm rates. In this regard, the most crucial consideration by experts is the improved feature contained dataset selection, such as UNSW_NV_15 datasets. Shushlevska et al. (2022) demonstrated that this dataset has remarkably responded when applied with Naive Bayes (NB) algorithms and shows the F1-score and Recall rates of 76.1% and 85.4%, respectively. On the other hand, for Logistic regression (LR), the estimated values are 78.2% & 96.1%, and for Decision Trees (DT), 88.3% & 95.4%; and the Random Forest, the scores are 89.3% & 98.5% respectively. In another study introduced by Vimal Rosy et al. (2021), different features from the UNSW-NV15 datasets are used based on which assaults and threats are detected from the network intrusion in users' computers. Herein, it has been reviewed that the author has selected an improved ML model - the "Validated Feature Selection-Estimated Classifier" (VFS-EC) model, which assessed these features, and the performance is measured to other existing models. As per the implication of the result, it can be stated the model has achieved a greater precision rate and accuracy result in the classification and detection of attacks than existing models. From this information, it is integral to consider that researchers have been verifying enhanced ML and advanced DL methods in recent years to determine novel attacks in users' network systems appropriately.

2.3 Deep Learning-based Methods for Identifying the Network Attacks in Cloud Environment

With appropriate consideration of the above information, it has been determined that machine learning models have been effective in predicting existing intrusion in the network system. However, the imbalanced characterization of datasets such as KDDCUP99, DARPA98, and NSL-KDD has shown specific limitations in the feature classification process with the trained ML models Sonule et al. (2020). Moreover, even though these classifiers were tested and trained with improved and well-balanced datasets like UNSW_NV_15, there is a marked ineffectiveness in the detection of novel attacks. Therefore, the research focus has been presented with a new integrated approach based on exploring the accuracy of deep learning algorithms in predicting novel attacks. In this review, the focus has been given to the classification and prediction accuracy of deep learning models using enhanced features from the UNSW_NV_15 dataset. In the study performed by Kamil and Mohammed (2023), the author explained that as an integral part of the reliable security system for network infrastructure, the "intrusion detection system" (IDS) has become a potential component that can prevent adversaries in intruding users' network, and detecting the traffic with modern attacks. However, in this approach, it has become a mandatory aspect to improve the IDS system to efficiently detect the attack. In this regard, a focus has been given to deep learning models - "Convolutional Neural Networks" (CNN) and "Dense Layers". In acknowledging the need to improve the effectiveness of these models, features from UNSW-NB15 datasets are used for preprocessing and training of the model. As per the evaluation of the model, it has been identified that the CNN architecture performed with a higher accuracy of 99.8% than the other model or existing models.

Another study presented by Ayantayo et al. (2023) has obtuse further knowledge on the remarkable success of deep learning models in the intrusion detection process in computer networks. However, as the author mentioned, the feature fusion impact is an underexplored area where features from two different datasets are combined to boost performance and the overall improvement of the generalized capability of DL models. In this regard, Ayantayo et al. (2023) employed features from the UNSW-NB15 dataset and NSL-KDD dataset. Although the approach is a new concept, the study has shown higher effectiveness regarding multi-classification tasks. Deep learning models have become a pro in the classification and detection process in the recent decade. However, the supremacy of these models still needs extensive research work to generalize their capability in terms of accuracy and precision level. With a focus on the challenging condition of network intrusion with increased cyber-attacks, substantial measures that can build potentiality in mitigating network threats are explored using this model architecture. As identified in the study by Dawoud et al. (2018), the detection of anomalies in the network system has deployed deep learning methods, particularly neural architectures, which have proven a successive approach with a performance accuracy of 99%. The explored study has identified the convenience of unsupervised and semi-supervised deep learning algorithms in detecting anomalies and has proven to be effective, especially autoencoders (a non-probabilistic DL algorithm) with the above-mentioned accuracy level.

The suitability of the CNN architecture has been explored by many researchers in their studies since this model is a "feed-forward" artificial neural network structure of DL models. As implored by Hooshmand and Hosahalli (2022), the "de facto standard" of the CNN model has been verified from significant operations, among which the detection of network anomalies has become a significant consideration. Moreover, on testing the model using the UNSW-NB15 network intrusion dataset, the F1-score obtained is 85%, 97%, 86%, and 78% for different features. Thus, it can be explained that the CNN model is highly efficient in the detection of network anomalies, especially when tested and trained with the UNSW-NB15 dataset Hooshmand and Hosahalli (2022). With exponential growth in the usability of the internet and cloud computing services, security threats in the network system have become a tremendous challenge. Moreover, in the recent decade, dynamism in the network threat with unknown attacks has denoted the inefficiency of conventional models. A study performed by Lin et al. (2019) has introduced an improved deep learning model - "long short-term memory" (LSTM) and "Attention Mechanism" (AM) as an extension to improve the accuracy performance. Based on the experimental result, it has been observed that the model serves an accuracy rate of 96.2%, thus signifying a high performance compared to most of the machine learning models. Hence, the overall understanding of the efficiency in terms of the accuracy level of DL models suggested the need to expand the research scope to explore the model's suitability further.

2.4 Other Potential Methods of Detecting Network Attacks in Cloud Computing Environment

In the above discussions, the contribution of information has extended knowledge on the single application of machine learning and deep learning models. However, this section has focused on hybrid classifiers and neural architectures that have proven their effectiveness

in classifying and predicting network intrusion using the UNSW-NB15 dataset Souhail et. al. (2019). Upon reviewing the study by Tahri et al. (2022), a hybrid machine learning model has been introduced whose application in network intrusion detection has been compared with the single models. The proven effectiveness of the hybrid model is further explored based on the performance of different parameters - confusion matrix, recall rates, accuracy level, precision rate, F1-score, specificity & sensitivity by using the UNSW-NB15 network intrusion dataset. The introduction and specification of hybrid models is a recent trend that has emerged to enhance the classification and detection process of different algorithms. In this regard, Dutta et al. (2021) explained that the application of DL methods has shown improved accuracy when used in the IDS system. However, it is often observed that with imbalanced datasets, the model accuracy is impacted, even in the case of DL methods. Therefore, a hybrid approach based on advanced deep learning - "Classical Autoencoder" (CA) and "Deep neural Network" (DNN) have been used by pre-processing with the UNSW-NB15 dataset. It has been identified that the model serves potential performance accuracy in intrusion detection compared to single models with the benchmarked dataset.

2.5 Novelty & Identified Gaps in existing Research

Previous research primarily utilized static models that had difficulty adapting to the fast-changing nature of cyber threats in cloud environments, often failing to recognize new and complex attack patterns. Existing anomaly detection systems in cloud computing often produced high false positive rates, reducing security team efficiency and leading to scenarios where real threats were potentially overlooked. While machine learning has been used in previous research, the integration of more advanced techniques like deep learning Autoencoders, and GNNs specifically for cloud security is less explored. This research introduces a novel approach by applying sophisticated Deep learning models like RNNs, Autoencoders, and GNNs specifically for cloud security. These models provide advanced capabilities in pattern recognition and anomaly detection, crucial for adapting to the evolving data patterns in cloud environments. This research stands out with its focus on real-time anomaly detection, aiming to develop a system that identifies and responds to threats as they happen, thereby significantly cutting down response times and potential damage, unlike traditional systems that rely on periodic analysis. We have integrated Deep learning models into a user-friendly, web-based interface, making advanced security monitoring accessible and promoting wider adoption across various user expertise levels.

Table 1: Comparison of Prior Work

Author	Framework	Approach	Advantage	Limitation
Lata and Singh (2022)	The framework involves analyzing existing security techniques in cloud computing, with a focus on Intrusion Detection Systems (IDS). It encompasses the evaluation of different service models (IaaS, PaaS, SaaS) and includes strategies like virtual machine introspection (VMI) and hypervisor introspection (HVI).	The research classifies IDS based on configuration, placement, and types of attacks detected. It emphasizes the importance of feature selection and dimensionality reduction in intrusion detection	Research offers a broad perspective on cloud security concerns, and the comparative analysis presents an in-depth understanding of various IDS methods and their effectiveness in different cloud service models.	Due to increasing complexity and variety of malicious activities, it is difficult to detect all attacks.
Attou et al. (2023)	Framework revolves around enhancing cloud security using a cloud-based intrusion detection model. The focus is on implementing and integrating machine learning algorithms, particularly the Random Forest (RF) classifier, for effective intrusion detection in cloud computing environments.	The approach involves utilizing Random Forest (RF) combined with feature engineering for intrusion detection. The model is trained and validated on two datasets, Bot-IoT and NSL-KDD	The proposed model demonstrates high accuracy (98.3% on Bot-IoT and 99.99% on NSL-KDD datasets), outperforming other methods like Deep Neural Networks (DNN), Decision Trees (DT), and Support Vector Machines (SVM).	Limitation is the lower recall with the NSL-KDD dataset, suggesting the need for further improvement in this area. Future work aims to address this by exploring Deep Learning (DL) and ensemble learning techniques to enhance the model's performance.
Bakro et al. (2023)	Framework focuses on developing an enhanced cloud intrusion detection system (IDS) that integrates advanced feature selection techniques and an ensemble model optimized by the Crow Search Algorithm (CSA).	The approach utilizes a blend of filter and automated models for enhanced feature selection and employs an ensemble classifier comprising LSTM, SVM, XGBoost, and FLN, with weights optimized by CSA. Tests are carried out multiple datasets.	System shows an efficient detection rate and low false alarm rate across different datasets and attack types, highlighting the robustness of the ensemble model.	Potential scalability issues with complex models, computational demands of ensemble methods, and the generalization of results across diverse and evolving cloud environments.

3 Methodology

Determining data points that exhibit substantially different properties from other portions of the dataset and correspond to normal occurrences is known as identifying anomalies. This method is becoming more and more important in several fields, however developing a reliable and autonomous anomaly identification process is difficult since there are few anomalous observations, a variety of patterns in anomalous data, and fluctuating operational circumstances while gathering data. Deep learning becomes a useful technique in automated anomaly detection since it helps identify extraordinary patterns, outliers, or exceptions. In this study, advanced algorithms are implemented to automate anomaly detection through a systematic set of procedures. The process begins with the collection of a legitimate dataset, followed by rigorous preprocessing. Comprehensive data analysis is conducted, including feature extraction and model training. The final phase involves evaluating models using various classification metrics to ensure an unbiased selection of the best-performing model. Each step in this workflow is elaborated upon in subsequent sections and the flow diagram of the entire workflow is shown in Figure 1.

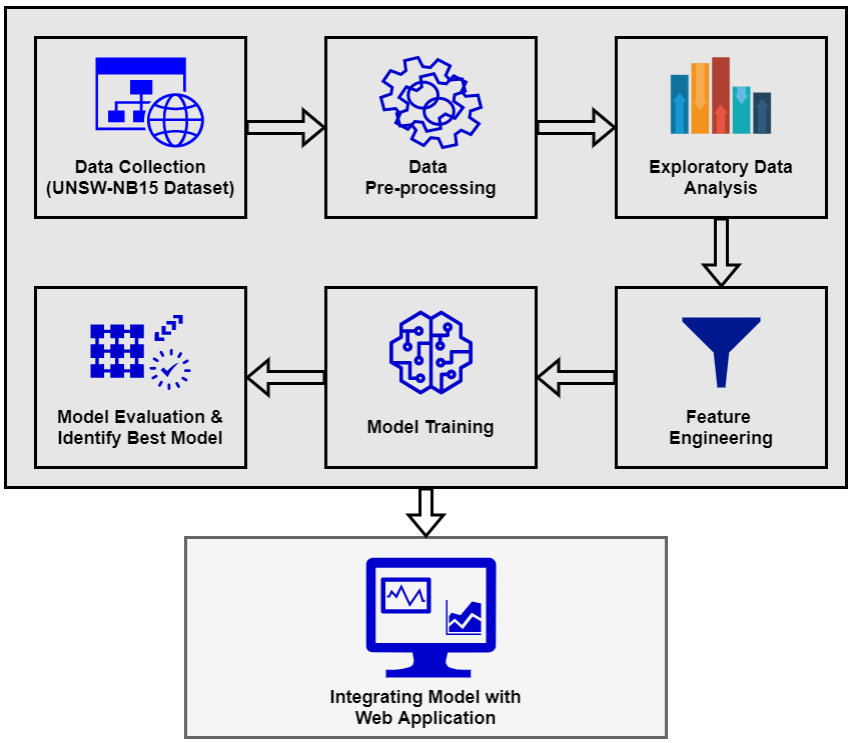

Figure 1: Methodology for Identifying Anomaly in Cloud Computing Network

3.1 Dataset Description

In the pursuit of constructing a comprehensive dataset for anomaly detection, data was meticulously gathered from the esteemed UNSW website, renowned for its credibility in providing anomaly data. The dataset, characterized by its voluminous nature, was systematically partitioned into four distinct CSV files to facilitate manageability. The initial phase involved assimilating pertinent features by referencing the 'Features File' as a mapping file, which encapsulates headers and their corresponding descriptions for the subsequent CSV files. The dataset manifests its extensive scale, with each of the

four files, boasting dimensions of (700000, 49), (700000, 49), (700000, 49), and (440043, 49), respectively. Following data retrieval, a pivotal step involved the visualization of the initial and terminal segments of each dataset, a prudent endeavor aimed at fostering a profound understanding of the data's inherent characteristics. This meticulous data collection and loading process lays a robust foundation for the ensuing stages of the anomaly detection study.

3.2 Data Preprocessing

After data collection the next step performed in this work is data preprocessing, a strategic approach was adopted to address the computational challenges posed by the expansive size of the dataset, requiring resource-intensive processing. To navigate this, a pragmatic solution was implemented, involving the concatenation of data from the four distinct CSV files into a unified structure. Subsequently, a meticulous strategy was employed to tackle the issue of class imbalance inherent in the dataset, wherein all rows corresponding to the minority class were extracted, and a random sample of rows from the majority class was meticulously selected. The resulting dataset, bore witness to a balanced distribution, comprising 700,000 rows and 49 columns. Stringent measures were taken to ensure data integrity, beginning with the removal of duplicate entries, culminating in a dataset dimension of 453,464 rows and 49 columns. An in-depth exploration of the dataset's composition ensued, encompassing a thorough examination of numerical and categorical features. Further refinements were implemented to rectify anomalies within specific columns. Noteworthy transformations included the replacement of erroneous values and the introduction of appropriate numerical representations. The 'attack_cat' column underwent meticulous treatment to address null values, replacing them with the categorical label 'normal.' Additionally, efforts were made to standardize and rectify discrepancies in the 'service' column by replacing '-' with 'none.' The preprocessing journey extended to the normalization of hexadecimal values within the 'sport' and 'dsport' columns, ensuring a consistent numerical format. A fundamental transformation involved the conversion of IP addresses from their conventional dotted-decimal form to a decimal representation, contributing to homogeneity in the dataset. Null value imputation was executed using the 'constant' strategy for specified columns, thereby mitigating missing data concerns. This comprehensive data preprocessing endeavor laid the groundwork for subsequent phases of the anomaly detection study, fostering a refined and standardized dataset ready for rigorous model training and analysis.

3.3 Exploratory Data Analysis

In exploratory data analysis (EDA) step, we examine the dataset through various statistical techniques and visualizations to understand its characteristics, identify patterns, and detect anomalies. All graphs and charts related to EDA, providing a comprehensive visual understanding of the dataset, are explained in detail in Chapter 5.

3.4 Feature Engineering

Feature Engineering is a pivotal stage in optimizing the dataset for deep learning applications, a series of transformations were implemented to enhance the efficacy of subsequent

model training. Various attack types (e.g., analysis, backdoor, DOS, exploits) were consolidated into a single 'malicious' category, simplifying the problem to normal versus malicious traffic. The categorical columns within the dataset were subjected to Label Encoding, a process crucial for translating categorical variables into numerical representations. Subsequently, addressing the challenge of class imbalance as shown in Figure 2, a Random UnderSampler was employed to equilibrate the distribution of the 'Label' variable, ensuring a more balanced representation of normal and anomalous instances as depicted in Figure 3. This step is instrumental in preventing model bias towards the majority class, thereby enhancing the robustness of the trained model. To facilitate efficient resource management and mitigate the risk of overfitting, dimensionality reduction was pursued through Principal Component Analysis (PCA). The significance of PCA lies in its ability to condense the dataset's complexity while retaining its essential information, surpassing other methods in terms of both efficiency and information preservation. This technique is particularly noteworthy for capturing the intrinsic patterns and variability within the dataset while reducing the number of features. The application of PCA resulted in a reduction to 18 components, preserving 98% of the information as illustrated in Figure 4. This not only contributes to computational efficiency but also mitigates the curse of dimensionality, enhancing the model's ability to generalize to unseen data. This streamlined and optimized feature set, achieved through Label Encoding and PCA, serves as a foundation for the subsequent stages of model training and evaluation in the context of the study.

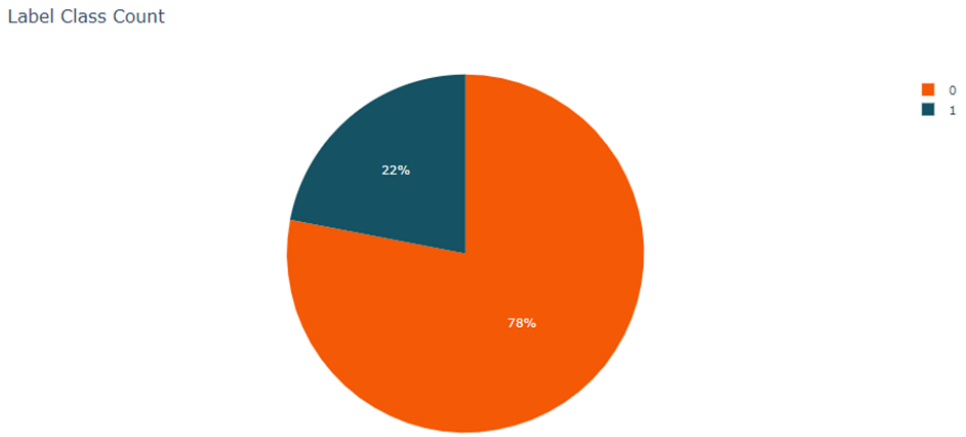

Figure 2: Imbalanced Class in Label Column

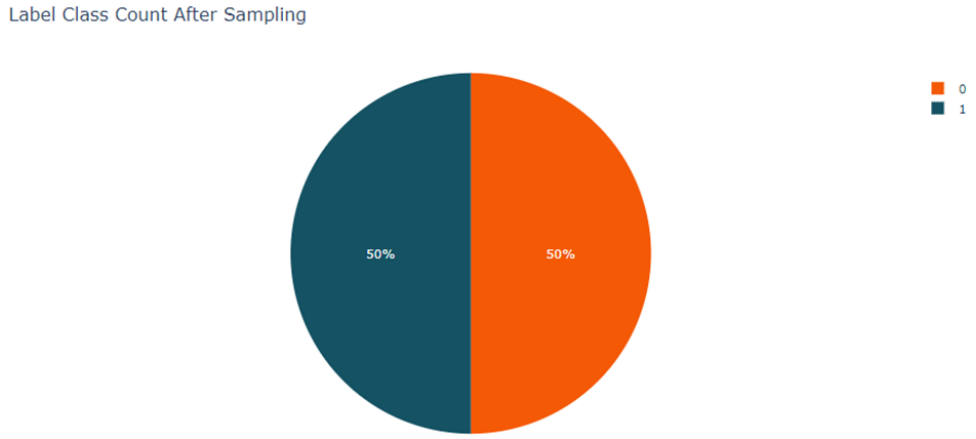

Figure 3: Balanced Class in Label Class After Sampling

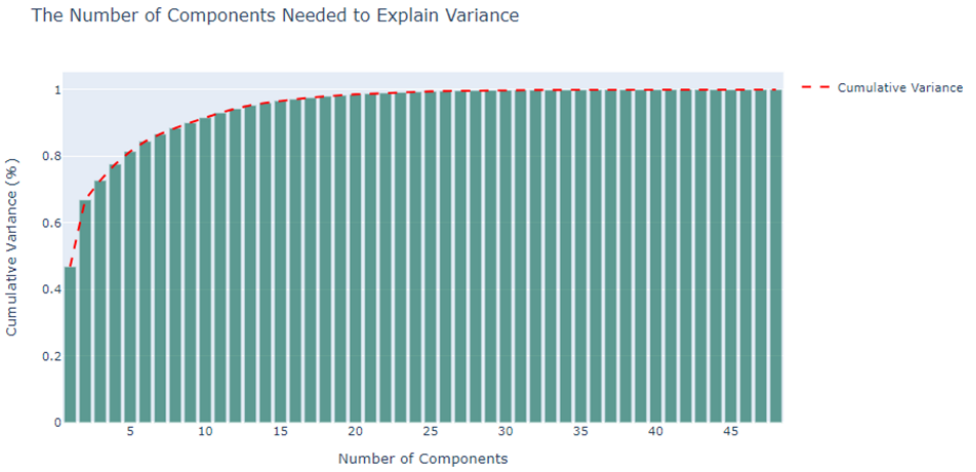

Figure 4: Principle Component Analysis (For Dimensionality Reduction)

3.5 Model Training

In this endeavor, a three-pronged approach was undertaken, implementing Recurrent Neural Networks (RNN), Autoencoder, and Graph Neural Networks (GNN) algorithms. The training data, shaped into a three-dimensional format, assumed a structure compatible with the requirements of algorithms, where the input shape consisted of a sequence of features. This reshaping ensured that the temporal relationships within the data were preserved during model training. The resulting dimensions of the reshaped training data were aligned with the specifications conducive to the Models' sequential learning capabilities. Each model was trained using binary cross-entropy loss and the Adam optimizer. The choice of binary cross-entropy as the loss function is pertinent to binary classification tasks and the Adam optimizer, renowned for its efficiency and effective model convergence. The training regimen involved iterating over the dataset for 10 epochs, with each epoch consisting of batch-wise updates, each batch comprising 512 samples. This approach, allows it to discern underlying patterns and relationships.

3.6 Model Evaluation

In the evaluation phase, the performance of each model was analyzed using four key metrics: accuracy, true positive rate (sensitivity), false positive rate, and specificity. All these metrics can be derived by calculating True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) values. These metrics provide a comprehensive assessment of the models' capabilities:

3.6.1 Accuracy

Measures the overall correctness of the model, indicating how often the model correctly identifies both normal and anomalous instances. Accuracy can be derived using the following formula:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

3.6.2 Sensitivity (True Positive Rate)

Evaluates the model's ability to correctly identify anomalous instances, reflecting its effectiveness in detecting actual threats. Sensitivity is also called as TPR (True Positive Rate). It can be derived using following formula.

$$\text{True Positive Rate (Sensitivity)} = \frac{TP}{TP+FN}$$

3.6.3 False Positive Rate (FPR)

Indicates the frequency of incorrectly classifying normal instances as anomalies, highlighting the model's potential for false alarms. The FPR can be derived using the following formula.

$$\text{False Positive Rate} = \frac{FP}{FP+TN}$$

3.6.4 Specificity

Assesses the model's accuracy in identifying normal instances, ensuring that non-threatening activities are not misclassified. Specificity can be derived using following formula.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

4 Design Specification

In this Chapter, we will discuss the architecture and working mechanism of all three deep learning algorithms Recurrent Neural Network (RNN), Autoencoder, and Graph Neural Network (GNN) used in our research. A description about each algorithm is provided in further subsections.

4.1 Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to process sequential data by incorporating the element of time into their architecture. Unlike traditional feedforward neural networks, RNNs possess internal memory that enables them to capture dependencies and patterns in sequential data. This makes them particularly effective for tasks such as natural language processing, speech recognition, and time-series analysis. RNNs operate by recursively applying the same set of weights to input data at each time step, allowing them to maintain a memory of past information. The working structure of this model is shown in Figure 5.

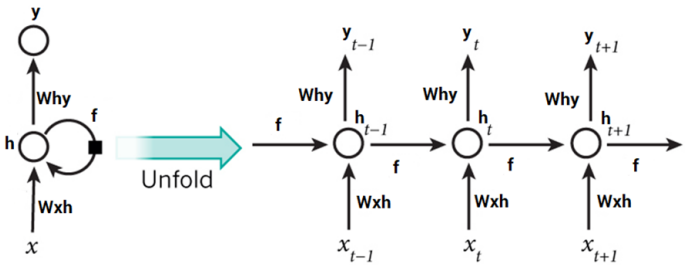

Figure 5: Design Architecture of Recurrent Neural Network (RNN) wu and Christofides (2019)

4.2 AutoEncoder

Autoencoders are a type of neural network architecture used for unsupervised learning, particularly in the domain of feature learning and data compression. Comprising an encoder and a decoder, an autoencoder aims to reconstruct input data at the output layer. The encoder compresses the input data into a latent space representation, and the decoder reconstructs the input from this representation. By training the autoencoder to minimize the reconstruction error, the latent space captures meaningful features within the data, serving as a condensed and informative representation. The working structure of this model is shown in Figure 6.

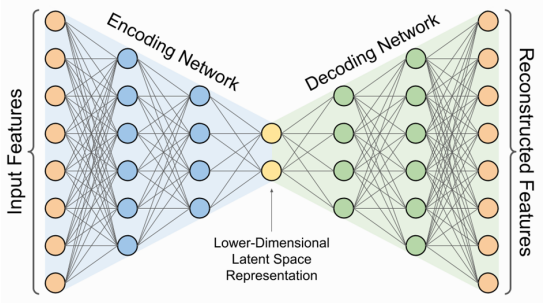

Figure 6: Design Architecture of Auto-encoder Petrov and Wortmann (2021)

4.3 Graph Neural Network (GNN)

Graph Neural Networks (GNNs) are designed to process and analyze graph-structured data, making them suitable for applications involving relational data, social networks, and molecular structures. GNNs operate by aggregating information from neighboring nodes in a graph, enabling them to capture complex relationships and dependencies within the data. This makes GNNs adept at tasks like node classification, link prediction, and graph classification, where understanding the intricate interactions between entities is crucial for accurate predictions. The working structure of this model is shown in Figure 7.

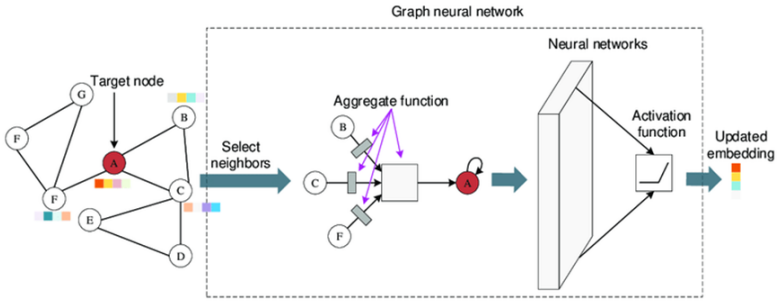

Figure 7: Design Architecture of Graph Neural Network Zeng and Tang (2021)

5 Implementation

In this section, we will be discussing about implementation of each step discussed in methodology and also we will discussing about the tool, technologies and different python libraries used for implementation.

5.1 Data Collection & Pre-processing

Data was collected from the UNSW website, known for its anomaly data. Python libraries like Pandas and Numpy for data handling and visualization. The initial steps are Reading, mapping, and understanding the dataset structure. For Data pre-processing We Concatenated CSV files, imputing missing values, and removing duplicates. Employing Sklearn's Random UnderSampler to address class imbalance. Converting hexadecimal values and transforming IP addresses for dataset integrity. Python libraries like Pandas and Numpy have been used to apply pre-processing.

5.2 Exploratory Data Analysis

Data analysis and visualization, a comprehensive exploration of various facets of the dataset was undertaken, illuminating key insights pertinent to the anomaly detection study. Python libraries such as Seaborn, Plotly, and Matplotlib for insightful data representations.

A visual depiction of the distribution of Source-to-Destination (S/D) and Destination-to-Source (D/S) transaction bytes, stratified by the 'Label' variable, was achieved through

pie plots. Figure 8 it is depicts that the sbytes label '1' constitutes its majority while for dbytes label '0' constitutes its majority.

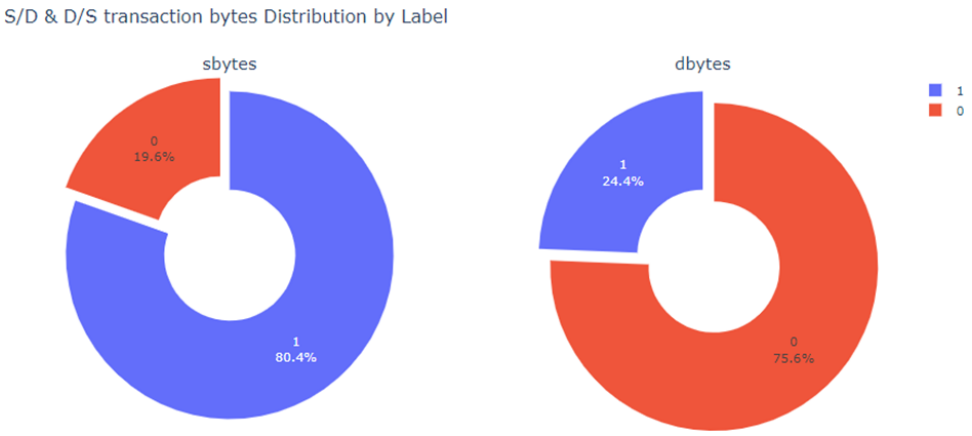

Figure 8: Source and Destination Bytes Distribution by Label

Further insights into the dataset's structure were gleaned through bar plots, particularly those focusing on columns with only two unique values. These visualizations, facilitated by Figure 9 and Figure 10 effectively elucidated the distribution patterns of binary features concerning the 'Label' variable.

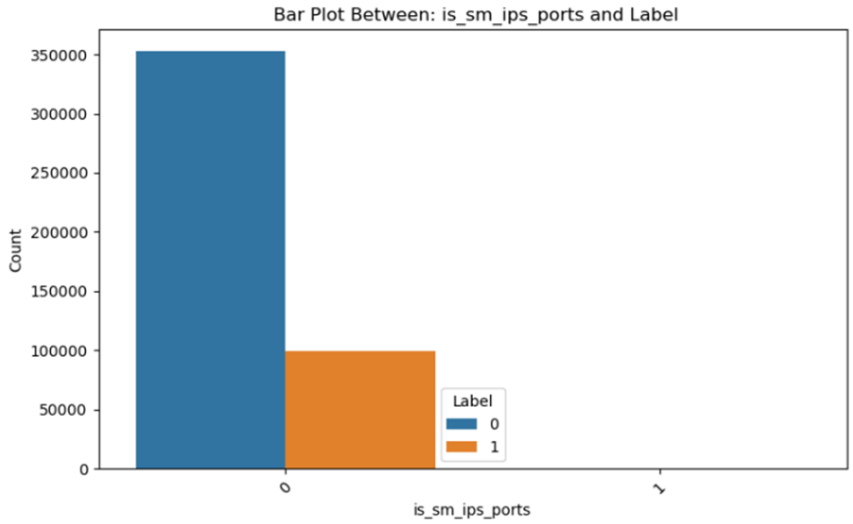

Figure 9: Bar Plot Between is_sm_ips and Label

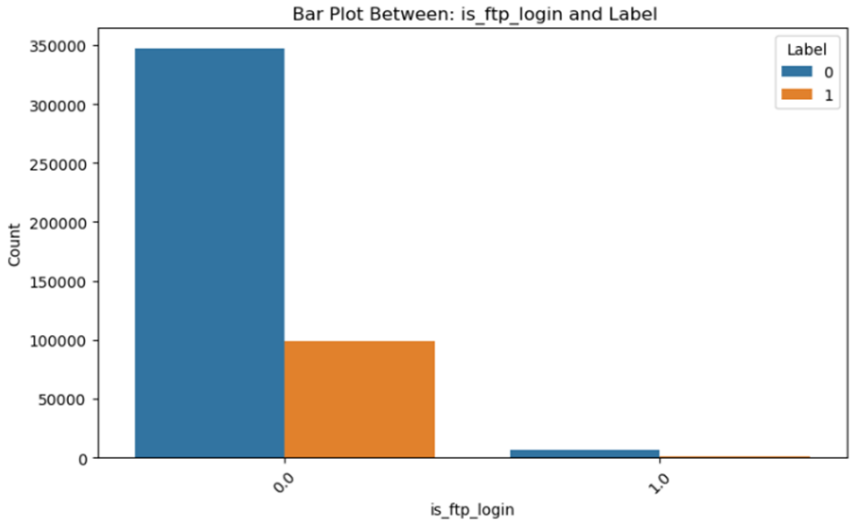

Figure 10: Bar Plot Between is_ftp$_{l}$*oginandLabel*

Additionally, a counterplot was employed to illustrate the diverse service counts concerning the 'Label,' shedding light on the prevalence of various services in normal and anomalous instances as depicted in Figure 11.

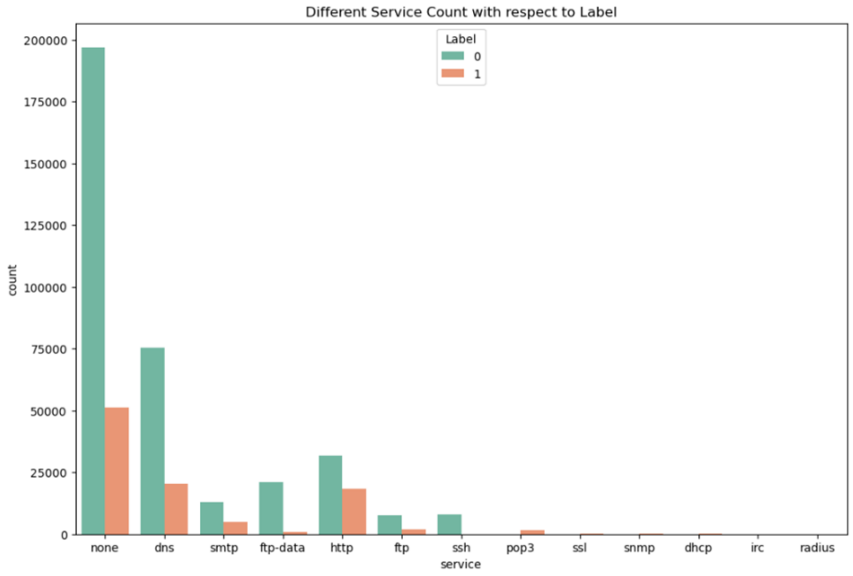

Figure 11: Different Service Counts concerning Label

The exploration extended to attack categories, visualized through a stacked bar plot showcasing the occurrences of each attack category about the 'Label' as shown in Figure 12. This elucidated the distribution of attacks across the dataset. A bubble scatter plot as depicted in Figure 13 was constructed to visualize the S/D packet count and D/S packet count, with the size of each bubble corresponding to the 'Label' variable, effectively highlighting the patterns associated with attacks.

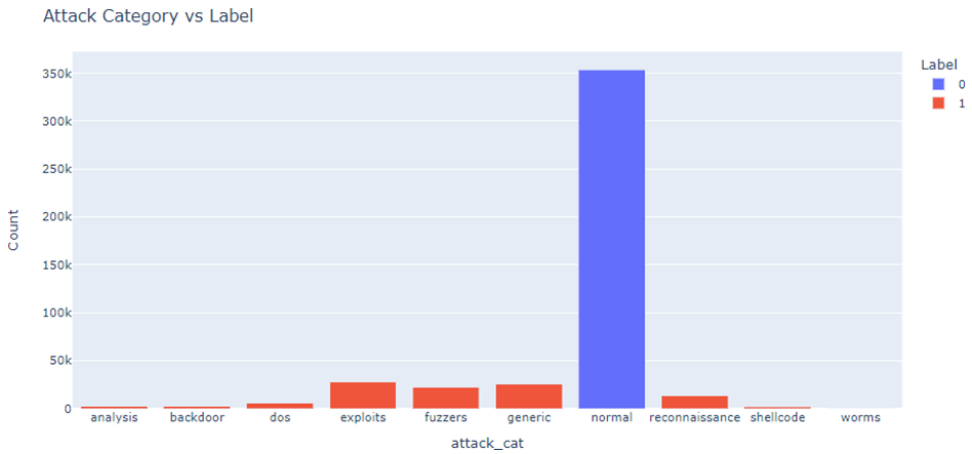

Figure 12: Different Attack Categories Wrt Label

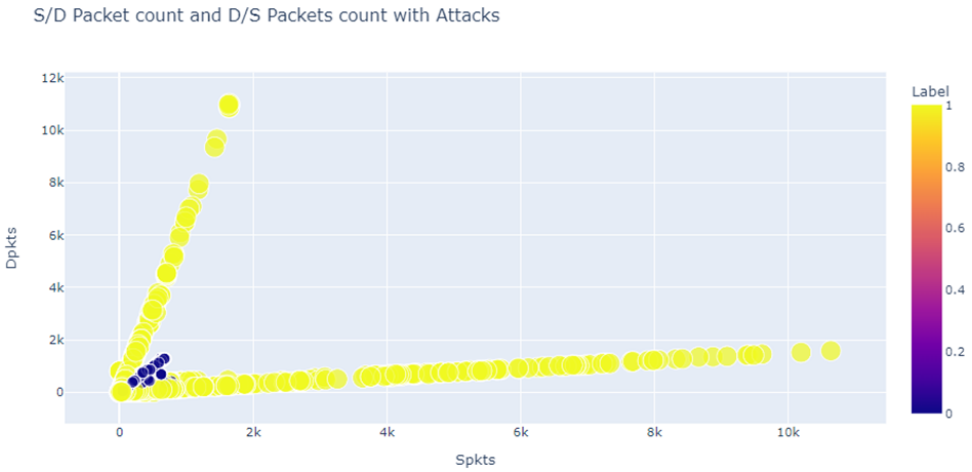

Figure 13: Scatter Plot between Spkts and Dpkts wrt Label

Histogram plots were leveraged to comprehend the data distribution of the 'sport' variable concerning the 'Label,' providing valuable insights into the frequency distribution of source ports in normal and anomalous instances as shown in Figure 14. The Source bits per second (Sload) and Destination bits per second (Dload) were also explored through a bubble scatter plot, shedding light on their distribution patterns concerning the 'Label' variable as depicted in Figure 15.

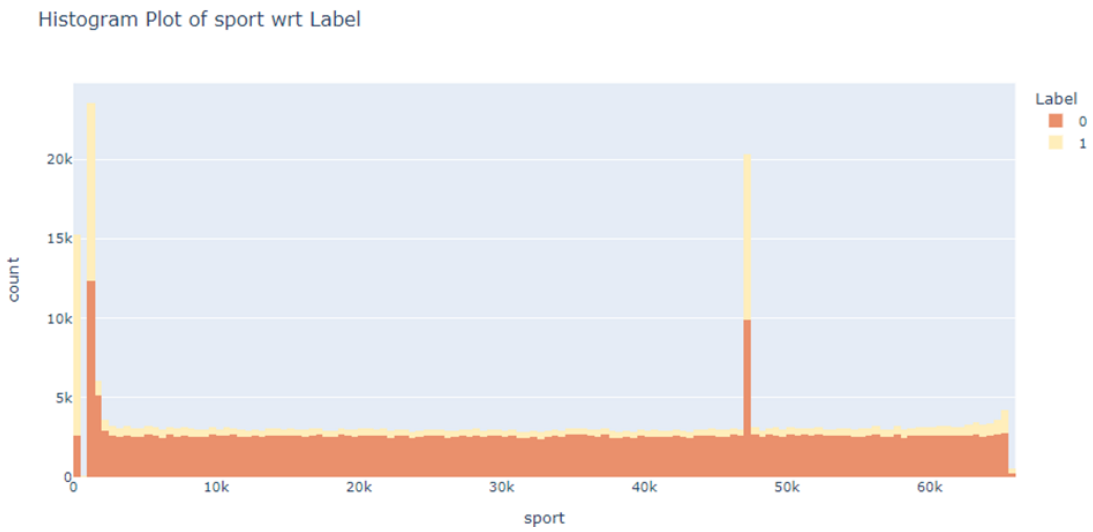

Figure 14: Histogram Plot of Sport column to the Label

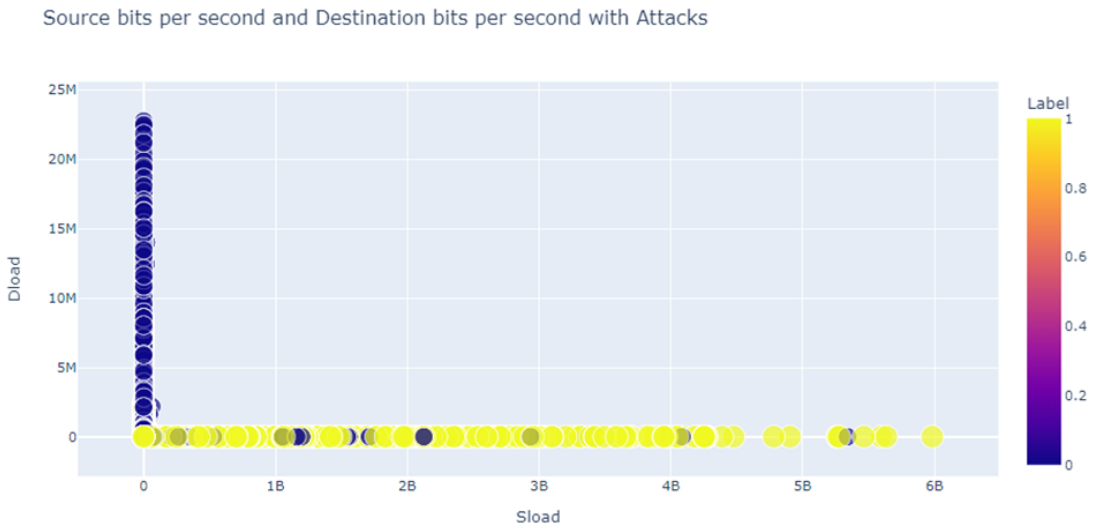

Figure 15: Scatter Plot of Sload vs Dload wrt Label

Finally, scatter plots were utilized to scrutinize the relationships between key variables, such as Source Arrival Time (Sintpkt) and Source bits per second (Sload) as shown in Figure 16, as well as Destination Arrival Time (Dintpkt) and Destination bits per second (Dload) as illustrated in Figure 17. These visualizations provided a holistic understanding of the interplay between temporal and bandwidth-related aspects in the context of normal and anomalous instances.

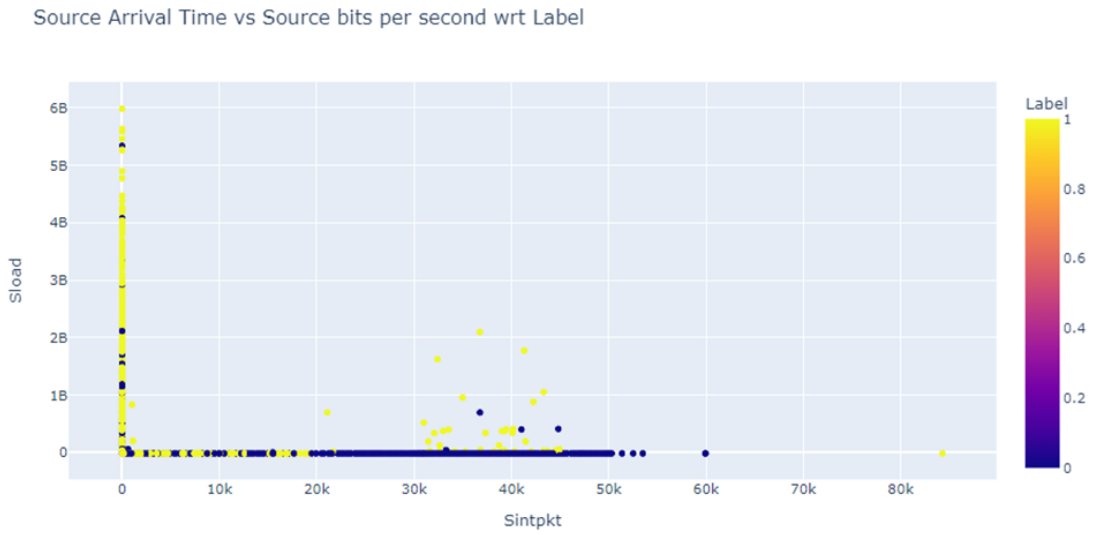

Figure 16: Scatter plot of Source Arrival Time vs Source Bits wrt label

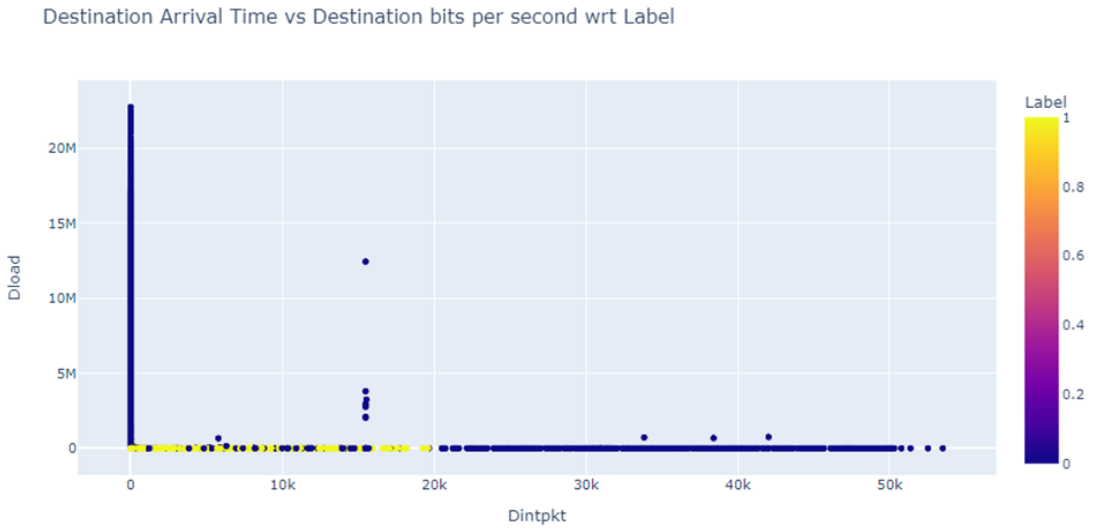

Figure 17: Scatter plot of Destination Arrival Time vs Destination Bits with label

These visualizations collectively contributed to a nuanced exploration of the dataset, paving the way for informed model training and analysis in the subsequent stages of the anomaly detection study.

5.3 Feature Engineering & Model Training

In this step, Converting categorical columns to numerical values and using PCA for dimensionality reduction was the main job. Python libraries used to implement are Scikit-learn and matplotlib. Tensorflow and Keras libraries are used for implementing deep learning models.

5.4 Web-UI Implementation

Monitoring network status in real-time is crucial for timely detection and response to anomalies or attacks. Therefore, a web interface has been developed for real-time network monitoring and anomaly detection. This interface serves as the primary user interaction point, providing real-time updates on network status and alerting users to potential anomalies. Snapshot of the Developed Web application has been shown in Figure 18 and Figure 19.

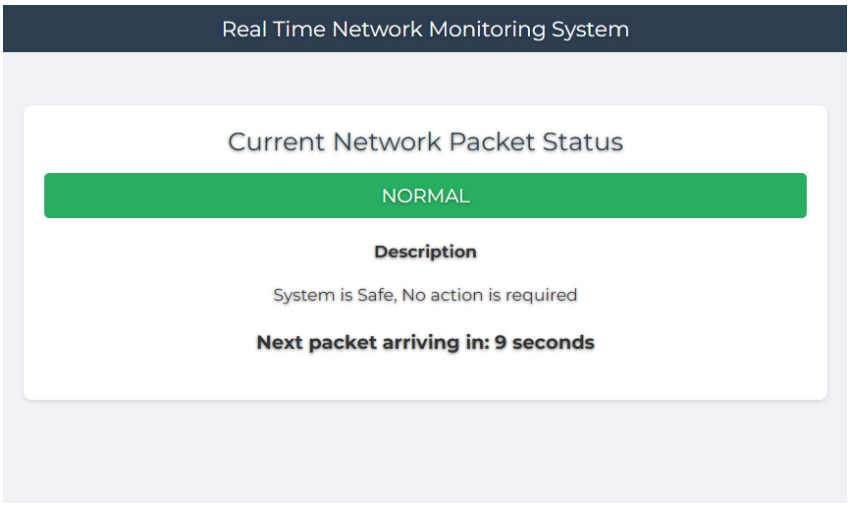

Figure 18: Networking Monitoring System Predicting Normal Traffic

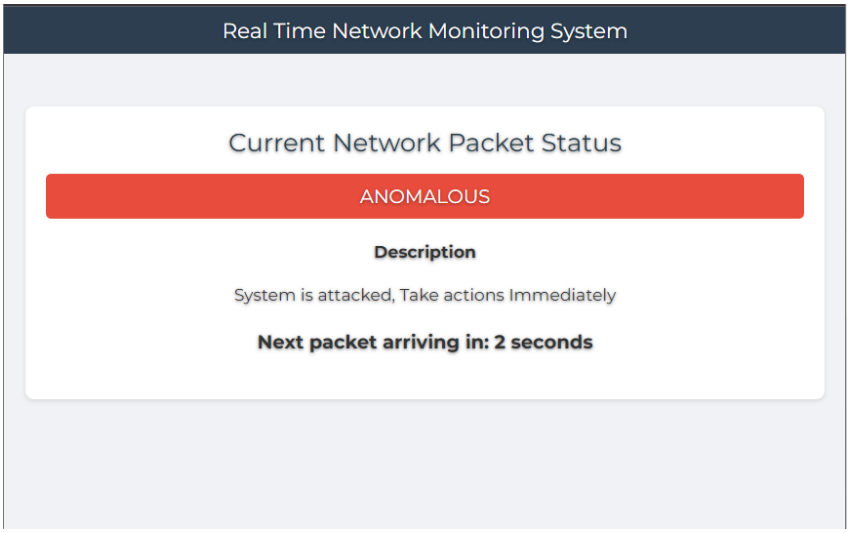

Figure 19: Network Monitoring System Predicting Anomalous Traffic

Key Components of our Web-UI are:

- **Flask Web Application:** The web application is built using Flask, a lightweight and efficient web framework in Python.

- **Integration with Model:** The application uses a Deep Learning Model to analyze network data. This integration allows for real-time prediction of network anomalies, classifying them as 'Normal' or 'Anomalous'.
- **Socket Communication:** Socket programming in Python was employed to facilitate real-time data communication between the server, client, and the network monitoring system. This real-time data flow is crucial for the timely detection of anomalies.

Architecture model
A web application was developed using Visual Studio Code and Python Flask, and deployed on AWS Cloud. An EC2 instance was set up with Ubuntu OS, and necessary libraries were installed. Application files were transferred from Flask to the EC2 instance, and connections to the instance were established using SSH or Putty. The application was operated by executing the 'UNSW-client.py' script in one terminal and running 'app.py' in another. Accessing the application via the URL "http://IP_address:5000" displayed the deep learning model's predictions on incoming packets, indicating 'Normal' traffic in green and 'Anomalous' traffic in red.

Table 2: Tools/Technologies Used

Tool Used	**Description/Specification**
Programming Language	Python
Cloud Environment	AWS
IDE	Visual Studio
Operating System	Ubuntu
Python Libraries	Pandas, Numpy, Scikit-Learn, Flask, Tensorflow, Keras

6 Evaluation

To ascertain the optimal approach for identifying confined connections and categorizing them into distinct types like anomalous or non-anomalous, it is imperative to conduct a thorough evaluation of classifiers based on crucial criteria. Detecting anomalies containing restricted records, connections, or associations proves more feasible. In this study, each methodology is assessed using key metrics such as accuracy, sensitivity, specificity, and AUC score. The selection of the best algorithm was driven by its adherence to the most stringent criteria set by these parameters in the context of the experimental data. The subsequent section delves into a detailed discussion of each algorithm's performance across various metrics.

6.1 Evaluation Based on Accuracy

The evaluation of the models based on the accuracy metric reveals notable distinctions in their performance. The Recurrent Neural Network (RNN) exhibited a commendable accuracy of 99.18%, signifying a high proportion of correct predictions. Autoencoder surpassed this performance, demonstrating an exceptional accuracy of 99.99%. The Graph

Neural Network (GNN) is closely followed with an accuracy of 99.98%. Notably, all models demonstrated an exceedingly high accuracy, underscoring their proficiency in correctly classifying instances. In terms of the best-performing algorithm, the Autoencoder stands out as the most accurate, achieving a near-perfect accuracy of 99.99%. This outcome suggests that the Autoencoder model excelled in capturing and reproducing the essential patterns within the dataset, leading to highly accurate predictions. The marginal differences in accuracy among the models highlight their overall effectiveness in learning and discerning between normal and anomalous instances. Similar accuracies across RNN, GNN, and Autoencoder models can be attributed to multiple factors. Firstly, the distinct features of the UNSW-NB15 dataset enable effective learning across different architectures, resulting in consistent high performance. Additionally, the sophistication of each model plays a role, as their inherent capabilities to capture complex patterns contribute to the observed accuracy levels. The comparison of Accuracy obtained by all models is illustrated in Figure 20.

Figure 20: Accuracy Comparison of Algorithms

6.2 Evaluation Based on Sensitivity

The second evaluation metric utilized for evaluation is Sensitivity, also known as True Positive Rate (TPR), a nuanced perspective emerges regarding the models' ability to correctly identify instances belonging to the positive class, specifically anomalies. The Recurrent Neural Network (RNN) exhibits an impressive Sensitivity of 0.9991, reflecting its high proficiency in accurately detecting instances of anomalies within the dataset. Autoencoder surpasses this performance, achieving a near-perfect Sensitivity of 0.9999, indicative of its exceptional capability to discern anomalies effectively. The Graph Neural Network (GNN) closely follows with a Sensitivity of 0.9998, aligning with its robust performance in capturing true positive instances. The Autoencoder, with its highest Sensitivity of 0.9999, stands out as the algorithm that excelled in correctly identifying anomalies. This result implies that the Autoencoder model demonstrated exceptional sensitivity to instances representing anomalies, effectively minimizing the occurrence of false negatives. The comparison of Sensitivity obtained by all models is illustrated in Figure 21.

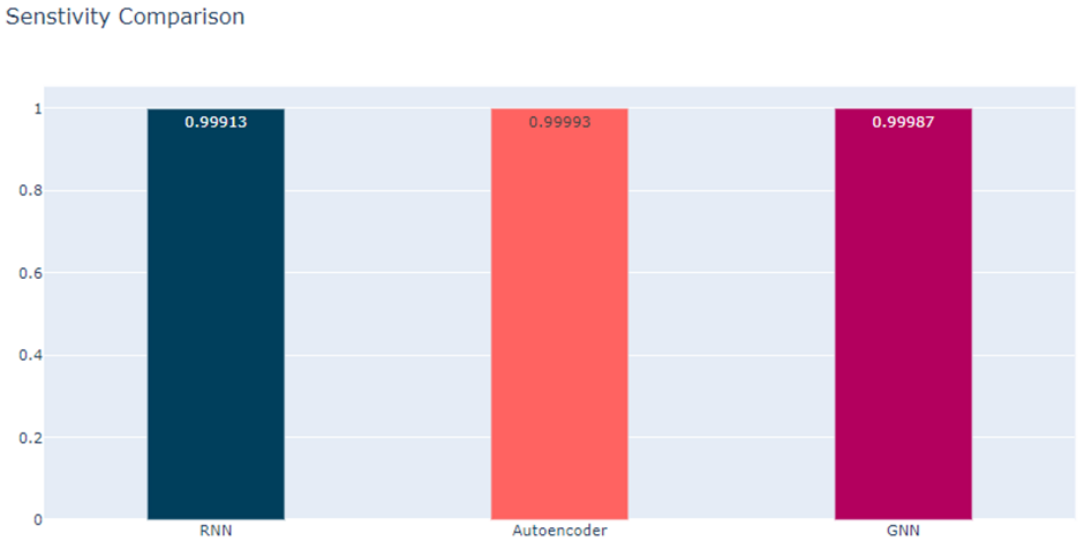

Figure 21: Sensitivity Comparison of Algorithms

6.3 Evaluation Based on False Positive Rate

In assessing the models based on False Positive Rate (FPR), a critical perspective emerges regarding their ability to minimize the misclassification of normal instances as anomalies. The Recurrent Neural Network (RNN) demonstrates a notably low False Positive Rate of 0.0155, indicative of its efficacy in avoiding the incorrect classification of normal instances. The Autoencoder outperforms this metric, showcasing an impressively minimal False Positive Rate of 0.0001, highlighting its proficiency in distinguishing normal instances from anomalies. The Graph Neural Network (GNN) closely follows with a False Positive Rate of 0.0002, aligning with its adeptness in minimizing false positives. The Autoencoder emerges as the algorithm with the most robust performance in minimizing false positives, recording the lowest False Positive Rate at 0.0001. The evaluation based on the False Positive Rate emphasizes the models' effectiveness in minimizing false positives, with the Autoencoder exhibiting the best performance and recording the lowest False Positive Rate. This outcome underscores the Autoencoder's proficiency in distinguishing normal instances, a crucial aspect of anomaly detection. The comparison of the False Positive Rate obtained by all models is illustrated in Figure 22.

Figure 22: False Positive Rate Comparison of Algorithms

6.4 Evaluation Based on Specificity

In the evaluation based on Specificity, a critical perspective emerges regarding the models' ability to accurately identify and classify normal instances, minimizing the occurrence of false positives. The Recurrent Neural Network (RNN) demonstrates a commendable Specificity of 0.9844, signifying its capability to effectively identify normal instances while maintaining a low rate of false positives. The Autoencoder surpasses this performance, showcasing an exceptional Specificity of 0.9999, highlighting its adeptness in precisely discerning normal instances. The Graph Neural Network (GNN) closely follows with a Specificity of 0.9977, aligning with its competence in minimizing false positives. This result underscores the Autoencoder's exceptional ability to accurately classify normal instances, thereby minimizing the likelihood of misclassifying them as anomalies. The comparison of the Specificity obtained by all models is illustrated in Figure 23.

Figure 23: Specificity Comparison of Algorithms

6.5 Evaluation Based on ROC-AUC Score

In the evaluation based on the Area Under the ROC Curve (AUC Score), a comprehensive perspective emerges regarding the models' overall discriminative ability and their performance across various thresholds. The Recurrent Neural Network (RNN) demonstrates a commendable AUC Score of 0.9967, indicating its effectiveness in distinguishing between normal and anomalous instances. The Autoencoder surpasses this performance, showcasing an exceptional AUC Score of 1, highlighting its superior discriminative capability. The Graph Neural Network (GNN) closely follows with an AUC Score of 0.9999, aligning with its proficiency in capturing intricate patterns indicative of anomalies. The Autoencoder stands out as the algorithm with the most robust performance, recording the highest AUC Score. The marginal differences among the models in the AUC Score underscore their overall competence in achieving high discriminative power. This result emphasizes the Autoencoder's superiority in capturing complex patterns, crucial for successful anomaly detection in the context of deep learning. The comparison of algorithms based on this metric is represented in Figure 24.

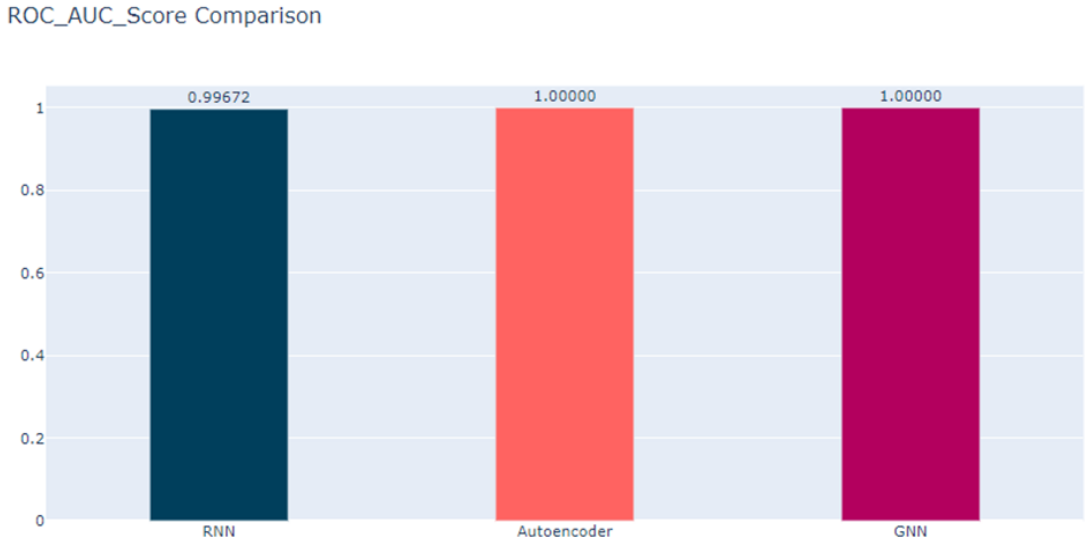

Figure 24: Roc_Auc Score Comparison of Algorithms

6.6 Discussion

The discussion of the study reveals compelling insights into the performance of the three implemented algorithms – Recurrent Neural Network (RNN), Autoencoder, and Graph Neural Network (GNN) – in the context of anomaly detection using deep learning. Each algorithm demonstrated commendable results across various evaluation metrics, underscoring their effectiveness in capturing intricate patterns and discerning between normal and anomalous instances. In evaluating the overall performance, the Autoencoder emerged as the algorithm that consistently outperformed its counterparts. It achieved the highest accuracy at 99.99%, followed by an outstanding Sensitivity of 0.9999. It also exhibited an impressively low False Positive Rate of 0.0001, emphasizing its capability to minimize the misclassification of normal instances. The exceptional performance of the Autoencoder is further substantiated by its highest Specificity at 0.9999 additionally, this model recorded the highest AUC Score at 1.0, signifying its superior discriminative

ability across different probability thresholds. While the Autoencoder emerged as the top-performing algorithm, it's crucial to acknowledge the commendable results achieved by the RNN and GNN. Both models demonstrated high accuracy, sensitivity, specificity, and AUC Score, showcasing their competence in anomaly detection. The superior performance of the Autoencoder can be attributed to its architecture's capability to learn and represent intricate patterns within the data, particularly those indicative of anomalies. The Autoencoder's ability to reconstruct input data while capturing essential features enables it to discern subtle variations, contributing to its robust performance. However, it's essential to recognize that the RNN and GNN also exhibited strong capabilities in anomaly detection, showcasing the versatility of deep learning approaches in handling complex tasks. The detailed analysis across multiple metrics provides a comprehensive understanding of each algorithm's strengths, collectively contributing to the successful implementation of anomaly detection methodologies in the given context.

7 Conclusion and Future Work

In this study, the application of deep learning algorithms for anomaly detection plays a pivotal role in strengthening cloud security and enhancing the safety of cloud services. Specifically, the use of Recurrent Neural Network (RNN), Autoencoder, and Graph Neural Network (GNN) has been pivotal in detecting complex patterns that are crucial for identifying threats in cloud computing environments. They can automatically learn hierarchical representations, which is crucial in identifying sophisticated attack patterns amidst normal traffic. Also, these models generally perform better as the amount of data increases. With large number of samples in our dataset, these models have shown improvement in accuracy and robustness. The standout performance of the Autoencoder, with its exceptional accuracy of 99.99%, demonstrates its potential as a robust tool for cloud security. Autoencoder's ability to minimize false alarms while accurately detecting real threats is invaluable in cloud security, where data integrity and system reliability are the major concerns. Although RNN and GNN also exhibited impressive capabilities. This study not only provides insights into the strengths of each algorithm but also serves as a guide for selecting the most appropriate models for cloud security applications. Additionally, the successful deployment of our web application on an Amazon EC2 instance marks a significant milestone in this research and showcases the practical implementation of our deep learning models for real-time attack detection in cloud environments. In the future, this research can be expanded to include security for IoT devices and edge computing, as these are increasingly integrated with cloud computing. The concept of a Hybrid Model can be explored to deal with more complex data and achieve higher accuracy to reduce false positive outcomes.

8 Presentation Demo Video Link

Link: `https://youtu.be/6qsOPcjP88M`

References

Attou, H., Guezzaz, A., Benkirane, S., Azrour, M. and Farhaoui, Y. (2023). Cloud-based intrusion detection approach using machine learning techniques, *Big Data Mining and Analytics* **6**(3): 311–320.

Ayantayo, A., Kaur, A., Kour, A., Schmoor, X., Shah, F., Vickers, I., Kearney, P. and Abdelsamea, M. M. (2023). Network intrusion detection using feature fusion with deep learning, *J. Big Data* **10**(1).

Bakro, M., Kumar, R. R., Alabrah, A. A., Ashraf, Z., Bisoy, S. K., Parveen, N., Khawatmi, S. and Abdelsalam, A. (2023). Efficient intrusion detection system in the cloud using fusion feature selection approaches and an ensemble classifier, *Electronics* **12**(11).
URL: *https://www.mdpi.com/2079-9292/12/11/2427*

Dawoud, A., Shahristani, S. and Raun, C. (2018). Deep learning for network anomalies detection, *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, IEEE.

Dutta, V., Choraś, M., Kozik, R. and Pawlicki, M. (2021). Hybrid model for improving the classification effectiveness of network intrusion detection, *13th International Conference on Computational Intelligence in Security for Information Systems (CISIS 2020)*, Advances in intelligent systems and computing, Springer International Publishing, Cham, pp. 405–414.

Hooshmand, M. K. and Hosahalli, D. (2022). Network anomaly detection using deep learning techniques, *CAAI Trans. Intell. Technol.* **7**(2): 228–243.

Kabir, M. H., Rajib, M. S., Rahman, A. S. M. T., Rahman, M. M. and Dey, S. K. (2022). Network intrusion detection using UNSW-NB15 dataset: Stacking machine learning based approach, *2022 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE)*, IEEE.

Kamil, W. F. and Mohammed, I. J. (2023). Deep learning model for intrusion detection system utilizing convolution neural network, *Open Eng.* **13**(1).

Kasongo, S. M. and Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset, *J. Big Data* **7**(1).

Lata, S. and Singh, D. (2022). Intrusion detection system in cloud environment: Literature survey future research directions, *International Journal of Information Management Data Insights* **2**(2): 100134.
URL: *https://www.sciencedirect.com/science/article/pii/S2667096822000775*

Lin, P., Ye, K. and Xu, C.-Z. (2019). Dynamic network anomaly detection system by using deep learning techniques, *Cloud Computing – CLOUD 2019*, Lecture notes in computer science, Springer International Publishing, Cham, pp. 161–176.

Moustafa, N. and Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set, *Inf. Secur. J. Glob. Perspect.* **25**(1-3): 18–31.

Nawir, M., Amir, A., Lynn, O. B., Yaakob, N. and Badlishah Ahmad, R. (2018). Performances of machine learning algorithms for binary classification of network anomaly detection system, *J. Phys. Conf. Ser.* **1018**: 012015.

Petrov, M. and Wortmann, T. (2021). Latent fitness landscapes -exploring performance within the latent space of post-optimization results.

Shushlevska, M., Efnusheva, D., Jakimovski, G. and Todorov, Z. (2022). Anomaly detection with various machine learning classification techniques over UNSW-NB15 dataset.

Sonule, A. R., Department of Computer Science & Engineering, Sir Padampat Singhania University(SPSU), Udaipur-313601, Rajasthan, India, Kalla, M., Jain, A., Chouhan, D. S., Department of Computer Science &Engineering, Sir Padampat Singhania University(SPSU), Udaipur-313601, Rajasthan, India, Department of Computer Science &Engineering, Sir Padampat Singhania University(SPSU), Udaipur-313601, Rajasthan, India and Department of Computer Science &Engineering, Sir Padampat Singhania University(SPSU), Udaipur-313601, Rajasthan, India (2020). Unsw-Nb15 dataset and machine learning based intrusion detection systems, *Int. J. Eng. Adv. Technol.* **9**(3): 2638–2648.

Souhail et. al., M. (2019). Network based intrusion detection using the UNSW-NB15 dataset, *Int. J. Comput. Digit. Syst.* **8**(5): 477–487.

Tahri, R., Jarrar, A., Lasbahani, A. and Balouki, Y. (2022). A comparative study of machine learning algorithms on the UNSW-NB 15 dataset, *ITM Web Conf.* **48**: 03002.

wu, Z. and Christofides, P. (2019). Economic machine-learning-based predictive control of nonlinear systems, *Mathematics* **7**: 494.

Zeng, Y. and Tang, J. (2021). Rlc-gnn: An improved deep architecture for spatial-based graph neural network with application to fraud detection, *Applied Sciences* **11**: 5656.