Enhancing Network Layer Security in Cloud Computing through Machine Learning Techniques

MSc Research Project
MSc in Cloud
Computing

Tejas Chavan
Student ID: X22206183

School of Computing
National College of Ireland

Supervisor: Aqeel Kazmi

National College of Ireland

MSc Project Submission

Sheet School of Computing

Student Name:	Tejas Chavan		
Student ID:	X22206183		
Programme:	MSc in Cloud Computing	**Year:**	2023-2024
Module:	MSc Research Project		
Supervisor:	Aqeel Kazmi		
Submission Due Date:	03/01/2025		
Project Title:	Enhancing Network Layer Security in Cloud Computing through Machine Learning Techniques		

Word Count: ………………………………………… **Page Count**………………………………………………..……..

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: Tejas Chavan

Date: 03-01-2025

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):	

Enhancing Network Layer Security in Cloud Computing through Machine Learning Techniques

Tejas
Chavan
X22206183

Abstract

In the rapidly evolving landscape of cloud computing, it is necessary to guarantee a robust network layer security system which protects the sensitive user data and thus maintains the integrity and availability of these cloud-based services. This research study explores the application of various advanced machine learning (ML) models in order to detect and reduce any critical vulnerabilities like Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access by using the CTU-13 dataset which contains relevant pipeline data including label consolidation, encoding, and balancing techniques. This dataset is used inside these ML models' training. These ML models include Logistic Regression, Random Forest, XGBoost, and Gradient Boosting Machine (GBM) and they are evaluated for their performance in classifying the network traffic and their ability to discover the malicious activities/patterns. Our results show that the ensemble-based models i.e. Random Forest, XGBoost, and GBM have significantly outperformed the baseline Logistic Regression model. These ensemble-based models, especially the Random Forest, achieved accuracy exceeding 99% and low False Positive Rates (FPR). These findings show the potential of such ML techniques in enhancing the security posture of these cloud environments i.e. for both individual and enterprise needs. In this research study, it is also discussed that there are trade-offs between model complexity and computational efficiency which gives better insights into the practical deployment of these models. This study concludes by pointing out the various key areas for future research like the integration of ML with blockchain and homomorphic encryption. This research contributes to the research community for the intelligent security frameworks in the cloud computing infrastructures which can handle the complex world of cyber-attacks.

Keywords: Cloud Computing, Network Security, Machine Learning, DDoS Attacks, Data Breaches, Unauthorized Access, Random Forest, XGBoost, Gradient Boosting Machine, Logistic Regression, CTU-13 Dataset.

1 Introduction

In this digital era the emergence of cloud computing has been a cornerstone of information technology infrastructure and this has revolutionized how both the individuals as well as enterprises manage to store and process their data. Such cloud computing services enable the businesses to enhance their operational efficiency by offering various scalable and on- demand resources to reduce their business costs. The adoption of these cloud services have been an important step for better data preservation and such services like Infrastructure as a Service, Platform as a Service, and Software as a Service, have been a critical part of finance, healthcare, education, and government data needs. These widespread cloud services are not without its challenges i.e. the need for complex security mechanisms to protect the user's sensitive data. Despite cloud computing's numerous advantages, a complex sum of such security challenges have risen naturally especially in the network layer. The network layer is responsible for data transfer between clients and cloud servers, thus making it an obvious target for malicious cyber attacks. This is due to the interconnected nature of these cloud environments and this makes the malicious actors to target the key vulnerabilities which are specific to such cloud network environments. These attacks include Distributed Denial of Service (DDoS), data breaches, and various unauthorized access attacks.

As the internet expands evermore, there is an increased frequency of cyber-attacks which target the cloud environments because of their role in modern day's data storage. This has been a continuous topic of the research community. Our study in this research paper dives deep into the advanced security measures like the integration of the ML models to detect the malicious patterns which work to target the security of cloud environments. The traditional security systems which often rely on static rule-based systems have now become increasingly unable to adapt to such new cyber threats and therefore there is a pressing need for such adaptive security frameworks like ours which can proactively detect and counter the cyber threats in real time by understanding the underlying malicious pattern.

In recent years, Machine Learning techniques have emerged as a promising tool in this context which we will be using in this research study to analyze vast amounts of data, identify both benign and malicious patterns, and thus make informed decisions with minimal human intervention. ML techniques with anomaly detection and classification have shown potential in enhancing the network security. This is done by enabling the automatic identification of the underlying malicious pattern which these ML techniques learn during their training. In this study such ML techniques are used to develop the necessary intelligent security systems that can not only respond to known threats but also predict unseen novel attack vectors with malicious patterns.

The transition to cloud-based infrastructures has not been without an increase in the cyber attack surface which makes it necessary to make strong counter security solutions. In this regard, the network layer is especially vulnerable. This is due to the network layer's role as data transmitter. DDoS attacks, Man-in-the-Middle attacks, and other such cyber attacks can overwhelm these cloud services. Data breaches and unauthorized access attacks leak user's sensitive information. In this regard, ML models can be tailored to counter these types of attacks but the traditional security solutions lack such ability. That is why this study proposes ML techniques to dynamically adapt to the evolving cyber threat inherent in such cloud environments.

The importance of this study is its potential to increase the security of network layers within cloud computing. This will safeguard the critical user data and ensure the uninterrupted operation of these cloud-based services. These findings will provide valuable insights into the application of ML in cloud security. This study also offers the foundation for future research of cloud-service security and the development of various complex security mechanisms.

1.1. Research Question

The primary objectives of this research are to comprehensively examine the predominant network layer vulnerabilities specific to cloud computing environments, including Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access. Additionally, the study aims to assess the effectiveness of various machine learning (ML) models—such as Logistic Regression, Random Forest, XGBoost, and Gradient Boosting Machines—in detecting and mitigating these identified network vulnerabilities. To achieve this, the research seeks to design and implement preprocessing techniques, including label consolidation, encoding, and balancing methods, that enhance the quality and suitability of network traffic data for ML model training. Furthermore, the study intends to determine the most appropriate performance metrics (accuracy, precision, recall, F1-score, False Positive Rate) for evaluating the efficacy of ML models in real-time threat detection. Beyond model evaluation, the research explores the scalability and adaptability of the proposed ML-based security solutions within dynamic and large-scale cloud environments, addressing challenges related to real-time threat detection and system performance. Finally, the study aims to identify key areas for future research, including the integration of ML with emerging technologies such as blockchain and homomorphic encryption, to further strengthen network layer security in cloud computing.

How can machine learning techniques be effectively employed to enhance network layer security in cloud computing environments by identifying, detecting, and mitigating specific cloud-specific vulnerabilities such as Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access?

To address this research question in this study, we will explore the following sub-questions:

- What are the predominant network layer vulnerabilities unique to cloud computing environments and which machine learning models (Logistic Regression, Random Forest, XGBoost, and Gradient Boosting Machines) are most effective in detecting and mitigating such network vulnerabilities?
- What preprocessing and data balancing techniques are necessary to prepare cloud network traffic data for machine learning model training, and what performance metrics (accuracy, precision, recall, F1-score, False Positive Rate) best gauge these machine learning models?
- How scalable and adaptable are the proposed machine learning-based security solutions in handling the dynamic and evolving nature of cyber threats in large-scale cloud environments and what are these challenges in the real-time threat detection using machine learning?

By addressing these sub-questions, this study provides a better and broader understanding of the role of machine learning in network layer security within cloud computing environments and the answers to these questions will also contribute in the development of these effective intelligent security systems which can deal with such complex cyber threats.

2. Related Work

In today's computing world, the cloud computing is a fundamental part of a vast portion of the internet computation and it has fundamentally changed the IT architecture by providing its users with the on-demand resources such as scalable computing, flexible loads, and cost effective techniques and models. This, however, has also introduced many challenges in the security department which has to deal with the cloud cyber security threats, especially the network layer which is prone to numerous cyber network attacks of varying degrees. To address such challenges various recent studies have focused on the well established Machine Learning techniques. This literature review spans its observation of such said ML techniques and how they can improve security, mitigate cyber threats, and counter any network based vulnerabilities which can pose a threat to the cloud environment.

Mukute et al. (2024) in a systematic study, conducted a review of the various security threats which have posed to the cloud computing environments over the years. Their study showed that a huge number of cloud security concerns revolved around data tampering, leakage, and unauthorized access. They also discussed various mitigation strategies against these attacks on the cloud computing environments including the encryption, access control, and usage/integration of the block chain itself in the core architecture of the cloud security. They showed that the cloud infrastructure had a fundamental flaw in its interconnected nature, which if not policed via various security solutions, could very well be a breaking point for the users safety and privacy. In his study, Mamushiane et al. (2023) explored these same challenges, especially the dynamic nature of the cyber threats in the cloud computing paradigm. His study presented a new angle into the analysis of the various vulnerabilities posed by cloud computing and how these gaps could be overcome by the usage of the various ML techniques. He showed that ML techniques when applied to the encrypted data and finding the patterns, of which it was already trained to find the malicious code, can be used to solve the various cloud computing cyber attacks without jeopardizing the users privacy. By doing such, it could be an effective path for future research to study the adaptive security measures on the cloud computing environments. Mamushiane et al. creative use of the various ML techniques could very well be the real-time cloud data protection which would solve both the static and dynamic security issues.

In his study of the cloud various services including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), Bonati et al. (2024) showed that there are many challenges faced by the various cloud service models. In their research, they categorized and focused on these issues for all individual network architectures by targeting each model itself. This was especially true for the Distributed Denial of Service (DDoS) attacks, and the unauthorized access via man-in-the-middle (MitM) attacks. Bonati et al. (2024) while following up on the various ML techniques to solve such network layer problems, also highlighted the critical role the network layer itself plays as a communication gateway to find and counter the various malicious actors which are bent on compromising the data integrity and/or data availability. Precursor to this study, Martin et al. (2023) already expanded on these concerns by ensuring that there was availability of the single or multi- cloud environments and their integrity layer. Martin et al. (2023) also highlighted the importance of the various intrusion detection systems alongside the prevention sub systems in the complete ecosystem of the cloud architecture. In his conclusion, Martin et al. (2023) argued that due to the ever changing nature of the various traffic patterns, the network layer must always be adaptive, always monitoring, and have enhanced traffic analysis to mitigate the threats effectively.

In his study of network layer security, Villa et al. (2024) discusses the various challenges posed by the smart city networks where cloud computing is used for the various data storage and processing tasks. Such networks are often prone to the various vulnerabilities which are unique to the cloud enabled smart city infrastructure which are faced with the various scalability issues and real-time traffic monitoring. This paper also discusses that the usage of the various ML techniques and real time traffic analysis was the way to solve these issues and make sure that the cloud security is deemed possible and made better with improved accuracy. Machine learning has emerged as a vast field and a proper technology for enhancement of cloud security. This is especially true for network layer vulnerabilities detection and their prevention. Linh et al. (2023) provided, in his research paper, a detailed review of ML algorithms used in cloud security. He categorized them into supervised, unsupervised, semi-supervised, and reinforcement learning while also noting that the supervised learning models, such as Random

Forest and Support Vector Machines (SVM), have been most effective in anomaly detection. He found out that consequently unsupervised models, like K-means clustering, have been well-suited for identifying unknown threats. In his research paper the authors stressed that the importance of feature selection and dataset quality in training are of paramount importance because these models perform best against the specific threats in cloud environments.

Chepkoech et al. (2023) introduced a novel cryptographic technique for cloud computing security. This technique was specifically used for highlighting the integration and the usage of ML with encryption methods i.e. enhancement of the data security. In this research paper, he proposes that the hybrid models when combined with the machine learning-based anomaly detection models alongside encryption layers, perform the best and ensure that even if network data is intercepted, it remains secure. This approach is closer to our own goals with the usage of various ML techniques that could be applied for real-time threat detection thus maintaining data integrity in turn through encryption.

In their paper, Lando et al. (2023) discussed the various technical challenges faced when implementing ML for cloud security. They showed in their research paper that by focusing on data privacy and the need for continuous learning models, various ML techniques are capable of adapting to newer threats in the cloud computing domain. In their research review, they highlight the current limitations in ML applications within cloud environments. These include computational resource requirements and the trade-off between detection accuracy and system performance. To mitigate such limitations, they suggest the need for optimized, resource-efficient models which can be used for the deployment in the cloud infrastructures.

While machine learning offers robust solutions for network layer security, it is also important to not diminish the impressive emerging technologies such as blockchain and homomorphic encryption. As they can be increasingly used to incorporate into cloud security frameworks. In this context, Routavaara et al. (2020) investigated the various roles of the blockchain in cloud security environments. He emphasized its potential to increase the data integrity and transparency in cloud transactions. He suggested that by maintaining unchanged and immutable records of data access, a single blockchain can be used to complement ML-based intrusion detection systems which in turn can be used for providing a secure audit trail which can help in the identification of the unauthorized access attacks. Following up on this study, Tykholaz et al. (2024) proposed his own with the emphasis on the homomorphic secret sharing as a promising approach to secure data in cloud environments. He discussed its potential for protecting sensitive data during computation in the multi cloud environment. His proposed methodology used the cryptographic method which allowed for the computations on encrypted data without even the need for the decryption which in turn could reduce the user's privacy concerns thus reducing network attacks. Tykholaz et al. argued in his research paper that integrating homomorphic encryption with ML-based intrusion detection could be very useful in the cloud computing environment itself and that it can provide a multi-layered approach to securing the network layer.

This raises anomaly detection and predictive analytics to the core of modern security frameworks that seek to secure cloud environments. Following work from Jahangeer et al. (2023) researchers such as themselves have suggested that the use of unsupervised learning models like K-Means clustering and Autoencoders could be used to learn and identify unusual network traffic patterns that can indicate attacks. And these are the methods that are good at finding new attack vectors that baseline systems, like rule based systems, or supervised models trained on historical data may not catch. Additionally, Alam et al. (2022) states that advanced ML tools such as Long Short Term Memory (LSTM) networks can aid predictability of attack patterns from current and historical data appearances. For example, such models for temporal data provide early DDoS attack patterns or prolonged brute force identification allowing a proactive layer of defence from potential attack. However, adopting these approaches inside the network layer security architecture allows these approaches to make an effective use from a broad perspective toward a multifaceted defense for both known and unknown threats. However, data preprocessing and feature engineering are still challenging effective steps in deploying machine learning models for cloud network security. The heterogeneity of cloud traffic data is so vast that Lando et al. (2023) mentioned that there are many difficulties: noise, missing values and high dimensions. Linh et al. (2023) extend this discussion and show how running feature selection techniques such as Recursive Feature Elimination (RFE), Principal Component Analysis (PCA) can help to reduce the computational overhead while keeping those features that are the most relevant to accurate anomaly detection. Nevertheless, according to Chepkoech et al. (2023), preprocessing should also address class imbalance since the presence of overrepresented benign traffic can tip out the training procedure and make it so that it cannot precisely detectals instances of malicious activity. Although these techniques like Random Under Sampling (RUS) and Synthetic Minority Over sampling (SMOTE) help but may in turn induce latent artifacts and consequently can cause the model to overfit the training data as well as under perform on real world data. Given that, research is needed in innovative preprocessing pipelines that adapt dynamically to the characteristics of incoming cloud traffic data towards more reliable and scalable ML-based security solutions.

Several studies identify ongoing challenges in cloud network security, especially pointing toward the need for adaptive and integrated solutions to the various challenges posed by the cloud environments. Sharma & Saxena (2020) in this regard, highlighted the lack of comprehensive solutions which could combine multiple technologies including various ML, blockchain, and encryption techniques i.e. to cover all the vast aspects of cloud security. Their survey suggested that there also exists a multi-pronged approach to solve the curious network layer vulnerability issue by creating standalone ML models or cryptographic measures which in turn may be used for the improvement against the cloud cyber attacks. Tan et al. (2023) and other researchers, continuing from there, also highlighted the importance of the encryption based techniques as most effective but limited without the various necessary support and supplementary methods. Such as machine learning to be used as a supplementary method for active threat detection. These many studies reinforce the importance of a layered security framework where ML-driven intrusion detection tries to complement the encryption and blockchain technologies present in the ecosystem of cloud computing which in turn is important for the creation of robust defense against these sophisticated network layer attacks.

3. Methodology

The primary objective of this research study is to increase the network layer security in cloud computing environments by using the machine learning (ML) techniques to identify, detect, and reduce the specific vulnerabilities like Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access. To achieve this research goal, we have used a comprehensive methodology and design encompassing data acquisition, preprocessing, model selection, training, evaluation, and deployment among the few. This methodology is designed to ensure that the ML models are both effective and efficient for the security of the network layer in the cloud computing environment.

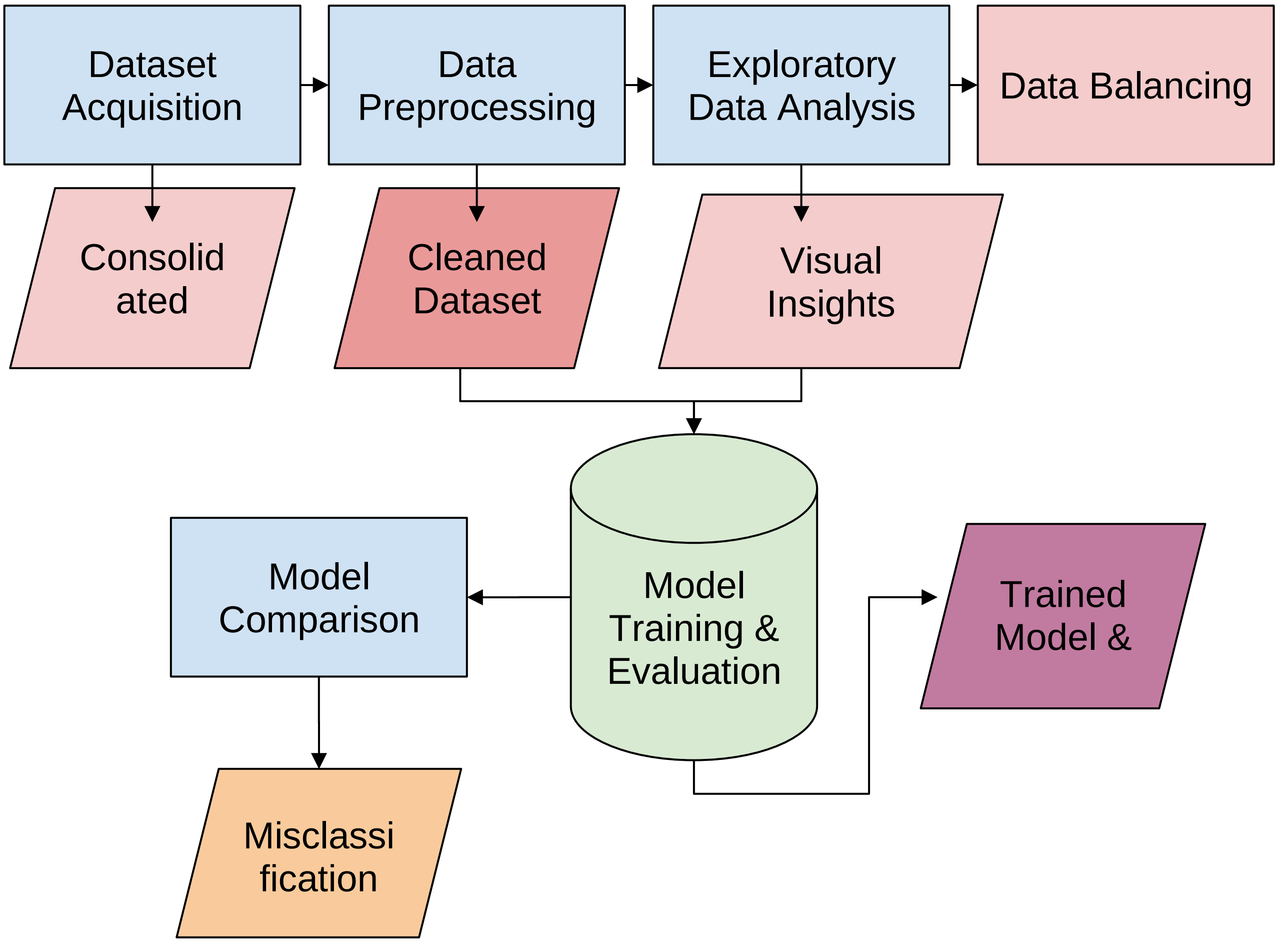

Figure 1: Overall Methodology workflow for the network layer security model training.

3.1. Data Collection

The CTU-13 dataset is chosen as the primary dataset for this study because this dataset is a widely recognized benchmark in the field of network intrusion detection. It consists of labeled network traffic data which is captured over various days with many attack scenarios. This dataset is especially suitable for this research because of its detailed nature i.e. it consists of multiple types of botnet traffic alongside normal network behavior. The CTU-13 dataset consists of 13 different scenarios where each scenario represents a unique botnet attack type alongside normal traffic and also that the dataset includes a variety of these features which act as network flows like duration, protocol, source and destination addresses and ports, traffic volumes, and labels for different type of traffic (e.g., normal, background, or specific botnet activities). For this research study we have gathered this dataset into a single DataFrame containing 19,976,700 entries across 15 columns which consists of both numerical and categorical variables.

3.2. Data Preprocessing

The dataset is acquired from the official CTU University of Prague repository. All the CSV files corresponding to the 13 scenarios are downloaded and used in a unified DataFrame using Python's pandas library where this combination also uses a complete analysis and preprocessing steps which are necessary for the ML model training in the later section of this research study.

3.2.1. Handling Missing Values

The initial examination of the CTU-13 dataset shows the presence of numerous missing values in several columns like in Sport, Dport, State, sTos, and dTos. These missing values are addressed by using a complete case analysis, where any rows containing null values are removed from the dataset and this is because of the vastness of the dataset. This approach makes sure that the data integrity is preserved and that it prevents the introduction of any new biases during the model training.

3.2.2. Feature Engineering

Two columns, StartTime and Dir, are thoroughly checked and declared as non-essential for the predictive modeling tasks and therefore they are dropped. This column dropping from the dataset is done to reduce the dimensionality and computational overhead. The StartTime column does not contribute anything major to the network layer security analysis, while the Dir (direction) column is considered to be redundant after comparison with the other traffic attributes.

3.2.3. Label Consolidation

The original Label column in the CTU-13 dataset contains specific traffic types. A custom function, categorize_label is used to merge these labels into three primary categories:

- Background Label (benign or routine network traffic).
- Botnet Label (all botnet-related activities, including DDoS attacks and unauthorized access).
- Normal Label (legitimate and non-malicious traffic).

This union/aggregation makes it easier for the classification task and it ensures that the ML models are trained on relevant important categories but the resulting Merged_Label column shows a significant class imbalance where the majority of entries in the column are classified as Background. This imbalance shows that there is a need for the application of various balancing techniques.

3.2.4. Encoding Categorical Variables

For the application of ML algorithms various categorical features like Proto, SrcAddr, Sport, DstAddr, Dport, and State are encoded into numerical representations. Label Encoding is used for this task which transforms these categorical string data into the integer labels that is helpful in the ML models training process. The class imbalance in the Merged_Label column is a serious issue and poses a significant challenge for the ML model training. To reduce this imbalance, a combination of Random Under-Sampling (RUS) and Synthetic Minority Over-sampling Technique (SMOTE) is used which is as follows:

- Random Under-Sampling (RUS) reduces the number of majority class (Background) instances to match the minority class (Botnet) by randomly removing samples from the majority class.

- SMOTE is applied to generate synthetic samples for the minority class (Botnet). This over-sampling method creates new instances based on the feature space similarities from the limited data.

3.3. Machine Learning Model Selection

The study evaluates the performance of four distinct ML models with unique strengths to tackle the various malicious attacks on the network layer security. This selection criteria for the ML models is based on ML model's individual strengths and uniqueness in handling high-dimensional and imbalanced datasets. These models are as follows:

- Logistic Regression is a fundamental classification algorithm that models the probability of a binary or multiclass outcome and this is entirely based on the input features. This model's simplicity makes it a valuable baseline model for intrusion detection tasks. Despite its linear nature, Logistic Regression can effectively capture relationships between features and target labels when appropriately regularized and tuned.
- Random Forest is an ensemble learning method which constructs multiple decision trees during training and then it outputs the mode of their predictions. This model's ability to handle large datasets with numerous features and also its great resistance to the overfitting makes it one of the best choices for detecting complex patterns in such network traffic dataset.
- XGBoost (Extreme Gradient Boosting) is a highly efficient implementation of traditional gradient boosting which is optimized for its speed and better performance than its counterpart. This model's ability to handle missing values, regularization to prevent overfitting, and parallel processing makes it a particularly suited model for such large-scale intrusion detection in this dataset.
- Gradient Boosting Machine (GBM) is another ensemble technique which is chosen because it builds models sequentially i.e. with each new model attempting to correct the errors of its predecessors. This makes it so that GBM's flexibility and high predictive accuracy is highly beneficial in the identification of the subtle anomalies for the network traffic data.

3.4. Experimental Setup

The experimental setup involves configuring the computational environment. Key configurations include:

- Ubuntu 20.04 LTS
- Python 3.8
- Pandas 1.3.5
- NumPy 1.21.5
- Scikit-learn 0.24.2
- XGBoost 1.5.2
- Imbalanced-learn 0.8.1
- Matplotlib 3.4.3

- Seaborn 0.11.2
- Joblib 1.1.0
- Tqdm 4.62.3

4. Design Specifications

In this section of the research study, we propose a design for a network layer security system for cloud computing environments. This proposed system integrates various components like data acquisition, preprocessing, machine learning modules, and security enforcement mechanisms.

4.1. System Architecture

Data Acquisition is the foundational step in this security framework which consists of collection of network traffic data from the cloud environments. In that regard, the CTU-13 dataset serves as the primary data source. It captures the diverse network traffic patterns like normal operations and various attack scenarios (DDoS attacks and botnet activities).

- Data preprocessing is critical to ensure the quality of the data for machine learning (ML) model training. This preprocessing step consists of several key steps:
    - Rows with null entries in critical columns (Sport, Dport, State, sTos, dTos) are removed to maintain data integrity.
    - Irrelevant or non-informative columns (StartTime, Dir) are discarded for dimensionality reduction.
    - The original Label column containing the granular classifications of network traffic is spread into three primary categories: Background, Botnet, and Normal.
- Categorical features (Proto, SrcAddr, Sport, DstAddr, Dport, State) are transformed into numerical representations using Label Encoding.
- Random Under-Sampling (RUS) is used to reduce the majority class (Background) instances to match the minority class (Botnet).
- Synthetic Minority Over-sampling Technique (SMOTE) is used to generate synthetic samples for the minority class to balance the dataset.
- StandardScaler is applied to numerical features to normalize their distributions.

4.1.1. Machine Learning Module

This Machine Learning Module is the core component of this research study. It is responsible for detecting and tackling the network layer vulnerabilities. This module consists of four distinct ML models.

- Logistic Regression
- Random Forest
- XGBoost (Extreme Gradient Boosting)
- Gradient Boosting Machine (GBM)

4.1.2. Overview of ML Models and Hyperparameters

Each ML model is trained using optimized hyperparameters to increase the performance. The following table provides an overview of key hyperparameters:

Model	Key Hyperparameters
Logistic Regression	C, penalty
Random Forest	n_estimators, max_depth, min_samples_split
XGBoost	learning_rate, max_depth, n_estimators, subsample
Gradient Boosting Machine (GBM)	learning_rate, n_estimators, max_depth, loss

4.2. Evaluation Framework

The Evaluation Framework is designed in such a way so as to assess the performance and quality of the implemented ML models. This framework consists of appropriate performance metrics, the setup of a controlled testing environment, and the execution of comparative analyses across different models. A comprehensive set of performance metrics is as follows:

- Accuracy (overall correctness of the model's predictions)
- Precision (proportion of true positive predictions among all positive predictions)
- Recall/Sensitivity (identification of all relevant positive instances)
- F1-Score (balance between precision and recall)
- False Positive Rate (rate at which non-threats are incorrectly classified as threats)
- Confusion Matrix (prediction outcomes across different classes)
- Detection Time (time taken by the model to process input data and generate predictions)

4.3. Evaluation Process Overview

The following table shows the evaluation process for each model:

Evaluation Step	Description	Tools/Techniques
Model Training	Training each ML model on the preprocessed and balanced training dataset	Scikit-learn, XGBoost
Hyperparameter Tuning	Optimizing model hyperparameters using Grid Search Cross-Validation	Scikit-learn GridSearchCV
Prediction and Inference	Generating predictions on the testing dataset	Scikit-learn, XGBoost
Performance Metrics	Computing accuracy, precision, recall, F1-score, FPR, and confusion matrices	Scikit-learn metrics

Calculation		
Visualization of Results	Creating confusion matrices and bar charts to visualize performance comparisons	Matplotlib, Seaborn, Scikit-learn ConfusionMatrixDispla y
Misclassification Analysis	Identifying and analyzing misclassified samples to understand model weaknesses	NumPy, Pandas

5. Implementation

The implementation phase of this research study consists of the practical execution of the above proposed methodology section i.e. creating a functional system designed to enhance network layer security in cloud computing environments. This section provides a detailed account of the steps which were taken including the environmental setup, data preprocessing, machine learning model training, deployment, and evaluation steps.

5.1. Environmental Setup

In order to implement the subject machine learning (ML) models and in order to deal with the large-scale datasets such as CTU-13, it is important to plan the environmental setup which in this case was carefully configured in order to attain the compatibility, efficiency, and reproducibility throughout the research. The following steps show the key components:

- These experiments were conducted on Ubuntu 20.04 LTS.
- Python 3.8 was used as the primary programming language due to its rich ecosystem of ML and data processing libraries.
- The research was carried out using Jupyter Notebook for interactive development and experimentation.
- Conda was used to create isolated environments as this practice reduces the potential conflicts.

5.2. Data Preprocessing

Effective data preprocessing is a crucial step for the performance of ML models and so the CTU-13 dataset required several preprocessing steps for training models. The CTU-13 dataset comprises multiple CSV files where they consist of different botnet attack scenarios. Using Python's Pandas library, all of these CSV files were loaded and then concatenated into a single DataFrame. Initial exploration showed that there were missing values in several columns (Sport, Dport, State, sTos, dTos). Rows containing any null values were removed using a complete case analysis approach and in order to streamline this dataset two columns i.e. StartTime and Dir were declared as non-essential and removed. This reduction in dimensionality not only enhances efficiency but also eliminates potential noise. The original Label column in the CTU-13 dataset contains granular classifications of network traffic. In that regard, a custom function was implemented to join all of these labels into three primary categories: Background, Botnet, and Normal. The resulting class distribution is as follows:

Merged_Label	Count
Background	17,485,027
Normal	351,907
Botnet	211,883

The above table shows that the Machine learning algorithms require numerical input and hence any categorical features are to be transformed into numerical using the Label Encoding technique. This transformation converts the string-based categorical data into integer labels, thus enabling these chosen ML models to process these features as ML models cannot deal with unchanged categorical data.

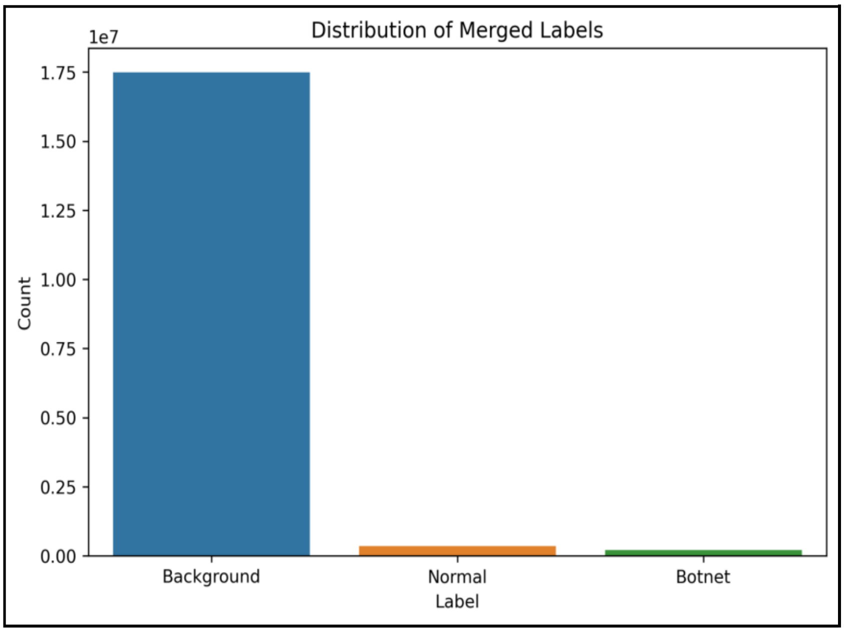

Figure 2: Distribution of Merged Labels showcasing the Normal & Botnet labels' imbalance.

5.3. Data Balancing

The Merged_Label column showed a significant class imbalance i.e. Background label dominating the entire dataset. To reduce and counter this imbalance a combination of Random Under-Sampling (RUS) and Synthetic Minority Over-sampling Technique (SMOTE) was used in this research study.

- Random Under-Sampling (RUS) reduced the number of Background instances to match the Botnet class size.
- SMOTE generated the synthetic samples required for the minority classes.

The balancing process was executed in batches and as such the final balanced dataset consists of 4,237,660 samples for each class (Background, Botnet, Normal), which is also shown below in tabular format:

Merged_Label	Count
Background	4,237,660
Botnet	4,237,660
Normal	4,237,660

This above table shows the balanced distribution which in turn ensures that ML models receive an unbiased training set to accurately detect all classes. To standardize the feature values of ML algorithms, StandardScaler was used for the numerical features which ended up scaling these values in order to ensure that all features have a mean of zero and a standard deviation of one.

5.4. Model Deployment and Saving

Post-training, each ML model was serialized and saved using Joblib. This was done in order to facilitate future deployment and for the use in the real-time inference applications. This serialization process makes sure that trained models can be used after training, can be loaded, and finally can be utilized without the need for retraining. The next steps involve the loading of these models in a production environment. This environment is then responsible for the integration of the real-time network monitoring on the incoming network traffic data.

6. Evaluation

This evaluation phase of our research study gauges the efficiency of the implemented machine learning (ML) models in enhancing the network layer security within cloud computing environments. These assessments were conducted using a predefined set of performance metrics, like Accuracy, Precision, Recall, F1-Score, False Positive Rate (FPR), and Detection Time. These metrics gauge each model's performance and provide a holistic evaluation of their abilities specifically to counter the vulnerabilities such as Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access. The following table summarizes the performance metrics for each ML model:

Model	Accuracy	Precision	Recall	F1-Score	FPR	Detection Time
Logistic Regression	70.20%	71.11%	70.20%	68.46%	12.33%	0.25
XGBoost	98.99%	98.99%	98.99%	98.99%	1.50%	2.07
Gradient Boosting Machine	99.41%	99.41%	99.41%	99.41%	0.49%	12.47
Random Forest	99.90%	99.90%	99.90%	99.90%	0.18%	49.92

The above table shows the comparative analysis of these ML models. It also shows their strengths and weaknesses which are inherent to each ML model alone. The table shows their suitability for enhancing the subject network layer security in the cloud computing environments. For the first ML model i.e. Logistic Regression showed moderate accuracy but higher FPR, thus indicating that while it can serve as a baseline, it is insufficient for our purposes of security applications in this network layer security. The following Confusion Matrix also shares this assessment for the Logistic Regression.

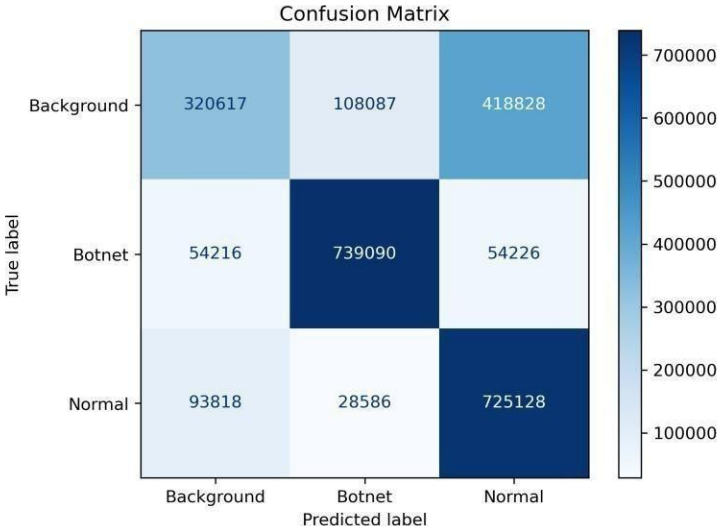

Figure 3: Confusion matrix for Logistic regression showcasing baseline model performance.

XGBoost emerges as a strong contender because of its high accuracy and low FPR. It also performed great when combined with relatively quick detection times thus making it a viable option for real-time threat detection. XGBoost's ability to manage missing values and its regularization techniques has contributed a lot to its results. The following figure shows XGBoost's performance:

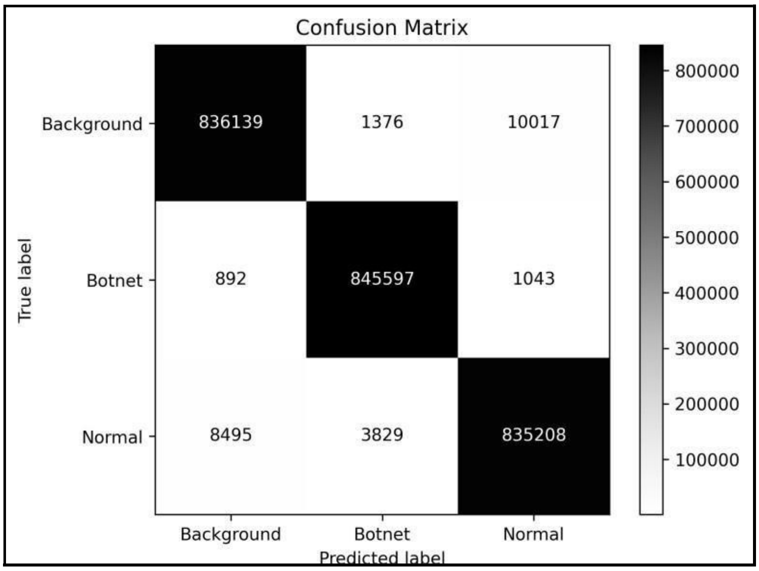

Figure 4: Better results shown by XGBoost performance against threat detection in real-time.

Gradient Boosting Machine (GBM) becomes easily the second best contender ML model as it further refines the performance achieved by XGBoost. This is because of the sequential nature of GBM which makes it iteratively improve its predictions thus reducing classification errors. However, the increase in detection time compared to XGBoost can be considered a trade-off between performance and computational efficiency and is thus dependent on the specific requirements and the limitations of environments requiring decision times to be faster.

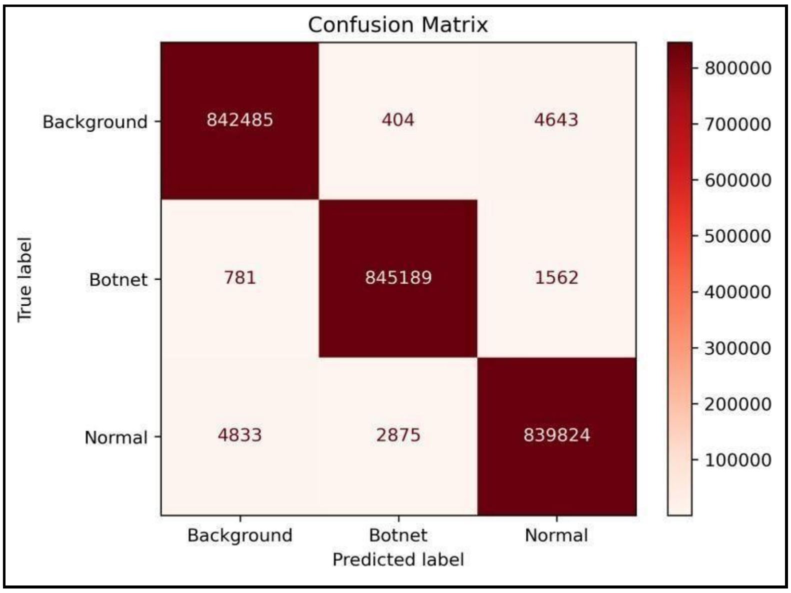

Figure 5: Confusion Matrix representing the GBM performance and superiority over XGBoost.

Random Forest stands out from the rest of the ML models which were chosen in this study with its exceptional performance metrics, achieving near-perfect accuracy and an almost negligible FPR. This is because of its ensemble approach which sums up the predictions of multiple decision trees. This method thus improves its generalizability. This also makes it so that it is highly reliable for diverse cloud environments. The primary drawback of Random Forest is its longer detection time which can be a serious trade-off in scenarios where ultra- fast response times are needed. This study concludes with Random Forest performance being the best and the following Confusion Matrix shows it clearly:

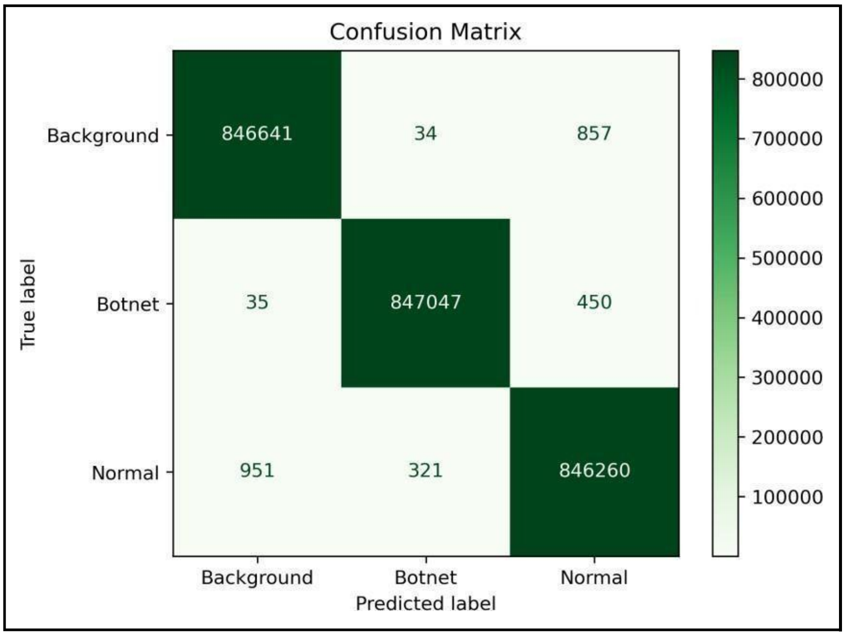

Figure 6: Best model performance from Random Forest with higher accuracies across all classified labels.

While ensemble models like XGBoost, GBM, and Random Forest performed better than Logistic Regression, computational resources and detection time must also be considered as a guide during their deployment. Logistic Regression is still a valuable ML baseline model for performance comparison for scenarios where speed is prioritized over higher accuracies. The following graph shows the comparative performance metrics of all of the four chosen ML models:

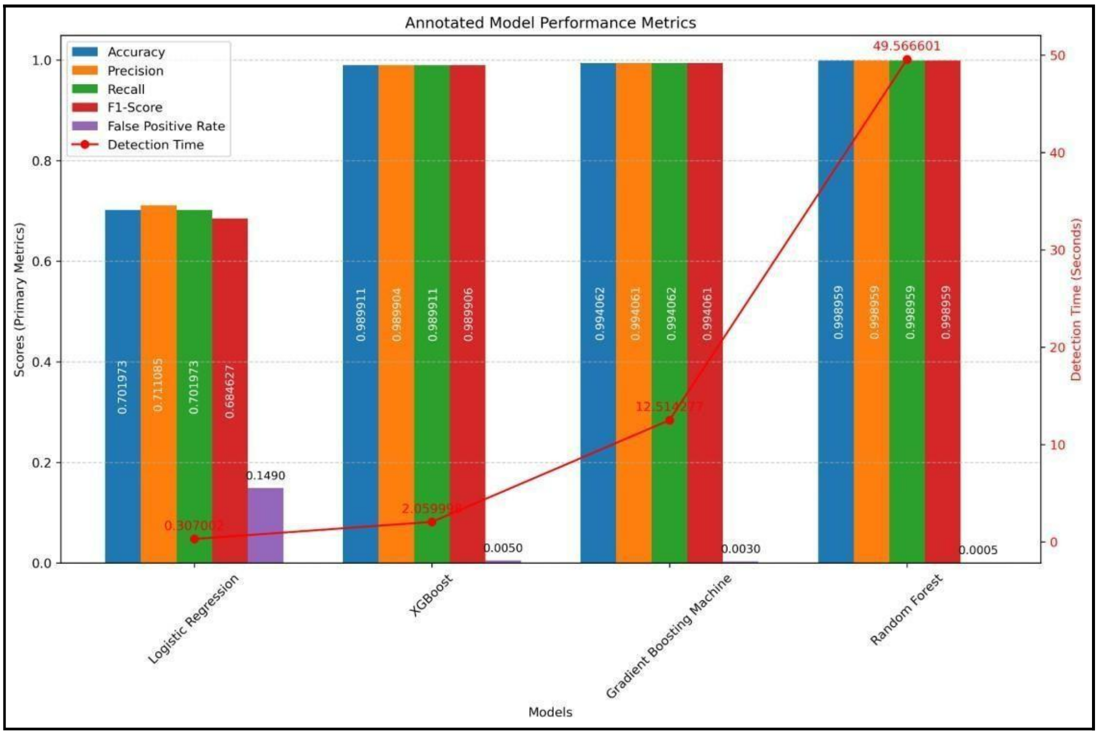

Figure 7: Model comparisons for all four models in a combined bar and line chart.

For the application in a real time network monitoring system, implementing machine learning models is not only model efficient but also affected by system architecture. This study demonstrates how trained models can be integrated with cloud native tools, like AWS CloudWatch and Azure Security Center, to continuously monitor network traffic. These platforms offer real time data streams to feed right into deployed models for anomaly detection. Lightweight deployment frameworks, like TensorFlow Serving or ONNX Runtime, are used to facilitate low latency and high throughput by handling the computational demands. In addition, the models built within containerized environments, such as Docker or Kubernetes, can be seamlessly scaled to meet fluctuating inbound traffic loads. To accommodate for this, the models are scalable dynamically so that even during high traffic situations (e.g., Distributed Denial of Service attack), the detection accuracy is not degraded much and processing speed is not reduced significantly. Real-time alerting mechanisms integrated with these frameworks further boost system responses in executing security teams by alerting them ahead of time about any such potential threats.

The great hurdle in building long term efficacy against evolving cyber threats is the robustness and adaptability of the deployed ML models. Another important implementation strategy uses continuous learning pipelines. They have these pipelines that will retrain these models on other data that they acquire all the time for these attack patterns and traffic behaviors that are coming up to make sure that their models are always up to date. Here we introduce techniques such as online learning and incremental model updates which help reduce downtime during the retraining without compromising good performance consistency. Additionally, these model interpretability tools, such as SHAP (SHapley Additive exPlanations), or LIME (Local Interpretable Model-agnostic Explanations), enable security teams to reason behind the model's predictions. The resulting transparency not only helps build trust in automated decision making but refines model features towards better performance. Ensemble models such as Random Forest and Gradient Boosting can be teamed up with adversarial training to prepare a system for coping with complex attack techniques that seek to exploit particular model shortcomings. These provide flexibility to ensure that the deployed solution is an effective defense at launch but also provides a robust defence as cyber threats continue to evolve.

7. Conclusion and Future Work

This research has successfully shown the potential of machine learning (ML) techniques in enhancing the network layer security within cloud computing environments by iteratively identifying and then addressing the cloud-specific vulnerabilities like Distributed Denial of Service (DDoS) attacks, data breaches, and unauthorized access. In this study we have created a complete system for using advanced ML models to detect and counter these threats. The implementation of this system including label consolidation, encoding, and balancing techniques, has ensured that these ML models were trained on a high-quality and preprocessed dataset. The evaluation of these four distinct ML models i.e. Logistic Regression, XGBoost, Gradient Boosting Machine (GBM), and Random Forest, showed that the ensemble-based models (Random Forest, XGBoost, and GBM) outperformed the baseline Logistic Regression model in terms of accuracy, precision, recall, F1-score, and False Positive Rate (FPR). These findings also show the potential of these ensemble models in capturing the various complex patterns within the network traffic data. Random Forest emerged as the top-performing model, achieving an exceptional accuracy of 99.90% and an F1-Score of 99.90%, coupled with a remarkably low FPR of 0.18%. Its ability to handle high- dimensional data and resist overfitting through its ensemble nature makes it an ideal candidate for deployment in dynamic and large-scale cloud environments. XGBoost and GBM also demonstrated superior performance metrics, highlighting their suitability for real- time threat detection tasks where a balance between speed and accuracy is essential. The study's findings not only validate the effectiveness of ML techniques in fortifying network layer security but also highlight the critical considerations in model selection and deployment. The trade-off between computational efficiency and performance metrics, as observed in the detection times of the evaluated models, provides valuable insights for practitioners seeking to implement real-time security solutions in cloud infrastructures. Future research can explore the integration of ML models with emerging technologies such as blockchain and homomorphic encryption. Blockchain can enhance data integrity and transparency, while homomorphic encryption can ensure data privacy by allowing computations on encrypted data. Combining these technologies with ML-based threat detection can create a more comprehensive and multi-layered security framework. By addressing these areas, future research can build upon the foundation established in this study, advancing the state of network layer security in cloud computing through innovative and integrated machine learning solutions. The continuous evolution of cyber threats necessitates ongoing research and development to ensure that security measures remain robust, adaptive, and effective in safeguarding cloud infrastructures against increasingly sophisticated attacks.

8. References

Alam, M.Z., Reegu, F., Dar, A.A. and Bhat, W.A., 2022, April. Recent privacy and security issues in internet of things network layer: a systematic review. In 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 1025-1031). IEEE.

Bhatt, S., 2024. Security and Compliance Considerations for Running SAP Systems on AWS. Journal of Sustainable Solutions, 1(4), pp.72-86.

Bonati, L., Polese, M., D'Oro, S., del Prever, P.B. and Melodia, T., 2024. 5G-CT: Automated deployment and over-the-air testing of end-to-end open radio access networks. IEEE Communications Magazine.

CHRISTOPHER, G., Joshi, K. and Patel, B., Navigating Data Protection Challenges in Amazon Web Services: Strategies and Solutions.

Chaganti, R., Gupta, D. and Vemprala, N., 2021. Intelligent network layer for cyber-physical systems security. International Journal of Smart Security Technologies (IJSST), 8(2), pp.42-58.

Chepkoech, M., Modroiu, E.R., Mwangama, J., Corici, M. and Magedanz, T., 2023, November. Evaluation of OSS-Enabled OpenRAN Compliant 5G StandAlone Campus Networks. In 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-7). IEEE.

Dai, Q.Y., Zhang, B. and Dong, S.Q., 2022. A DDoS-Attack Detection Method Oriented to the Blockchain Network Layer. Security and Communication Networks, 2022(1), p.5692820.

Dastres, R. and Soori, M., 2021. A review in recent development of network threats and security measures. International Journal of Information Sciences and Computer Engineering.

Fadhil, S.A., 2021. Internet of Things security threats and key technologies. Journal of Discrete Mathematical Sciences and Cryptography, 24(7), pp.1951-1957.

Fagan, M., Marron, J., Watrobski, P., Souppaya, M., Barker, W., Deane, C., Klosterman, J., Rearick, C., Mulugeta, B., Symington, S. and Harkins, D., 2023. Trusted Internet of Things (IoT) device network-layer onboarding and lifecycle management: Enhancing internet protocol-based IoT device and network security (No. NIST Special Publication (SP) 1800-36 (Withdrawn)). National Institute of Standards and Technology.

Jahangeer, A., Bazai, S.U., Aslam, S., Marjan, S., Anas, M. and Hashemi, S.H., 2023. A review on the security of IoT networks: From network layer's perspective. IEEE Access, 11, pp.71073-71087.

Jangjou, M. and Sohrabi, M.K., 2022. A comprehensive survey on security challenges in different network layers in cloud computing. Archives of Computational Methods in Engineering, 29(6), pp.3587-3608.

Lando, G., Schierholt, L.A.F., Milesi, M.P. and Wickboldt, J.A., 2023, May. Evaluating the performance of open source software implementations of the 5g network core. In NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium (pp. 1-7). IEEE.

Linh, A.B.N., Rupprecht, D., Poll, E. and Kohls, K., 2023. Analysing open-source 5G core networks for TLS vulnerabilities and 3GPP compliance.

Mamushiane, L., Lysko, A., Kobo, H. and Mwangama, J., 2023, August. Deploying a stable 5G SA testbed using srsRAN and Open5GS: UE integration and troubleshooting towards network slicing. In 2023 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD) (pp. 1-10). IEEE.

Martin, A., Losada, P., Fernández, C., Zorrilla, M., Fernandez, Z., Gabilondo, A., Uriol, J., Mogollon, F., Serón, M., Dalgitsis, M. and Viola, R., 2023. Open-VERSO: a vision of 5G experimentation infrastructures, hurdles and challenges. arXiv preprint arXiv:2308.14532.

Mukute, T., Mamushiane, L., Lysko, A.A., Modroiu, R., Magedanz, T. and Mwangama, J., 2024. Control Plane Performance Benchmarking and Feature Analysis of Popular Open-Source 5G Core Networks: OpenAirInterface, Open5GS, and free5GC. IEEE Access.

Routavaara, I., 2020. Security monitoring in AWS public cloud.

Sharma, P. and Saxena, R., 2020. Security Best Practices in AWS. NeuroQuantology, 18(8), p.389.

Tan, K.H., 2023. Mitigating Insider Threats in AWS: A Zero Trust Perspective.

Tykholaz, D., Banakh, R., Mychuda, L., Piskozub, A. and Kyrychok, R., 2024. Incident response with AWS detective controls.

Villa, D., Khan, I., Kaltenberger, F., Hedberg, N., da Silva, R.S., Maxenti, S., Bonati, L., Kelkar, A., Dick, C., Baena, E. and Jornet, J.M., 2024. X5G: An Open, Programmable, Multi-vendor, End-to-end, Private 5G O-RAN Testbed with NVIDIA ARC and OpenAirInterface. arXiv preprint arXiv:2406.15935.

Väisänen, T., 2023. Security review of Cloud Application architectures.