National College of Ireland

A Cloud-Based Network Intrusion Detection System Using Hybrid Ensemble Machine Learning and Explainable AI

MSc Research Project

Bhargavi Gadhepally
Student ID: x23344440

Masters of Science in Cloud Computing
National College of Ireland

Supervisor: Sai Guna Ranjan Emani

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Bhargavi Gadhepally

Student ID: X23344440

Programme: MSc in Cloud Computing **Year:** 2026

Module: MSc in Cloud Computing Research Project

Supervisor: Sai Guna Ranjan Emani

Submission Due Date: 28/01/2026

Project Title: A Cloud-Based Network Intrusion Detection System Using Hybrid Ensemble Machine Learning and Explainable AI

Word Count: 7578 **Page Count** 22

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: Bhargavi

Date: 28/01/2026

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):	

A Cloud-Based Network Intrusion Detection System Using Hybrid Ensemble Machine Learning and Explainable AI

Abstract

Cloud computing has increased the complexity and scale of modern network infrastructures by making them more vulnerable to complex cyber-attacks that traditional rule-based intrusion detection systems fail to detect. This study solves the challenge of achieving accurate, low-latency and interpretable network intrusion detection in cloud environments. The main objective of this study is to design and evaluate a cloud-based Network Intrusion Detection System (NIDS) that improves attack classification accuracy while maintaining transparency and practical deployment. To achieve this a hybrid ensemble machine learning approach where XGBoost as a meta-classifier. The system has been trained and evaluated with SMOTE which applied to reduce class imbalance. Explainable AI has been integrated using LIME to provide local, feature-level explanations for intrusion predictions. The proposed system has been implemented and evaluated in an AWS cloud environment by considering both detection performance and operational metrics. Experimental results has been shown that the hybrid ensemble model outperforms individual classifiers by achieving 90% accuracy and macro F1-score while handling overlapping attack patterns. The use of LIME enhances model interpretability by aligning with current state-of-the-art trends in explainable security analytics. Future work will focus on real-time traffic analysis and further optimization for edge-cloud environments.

Keywords: Network Intrusion Detection, Hybrid Ensemble Learning, Cloud, Latency

1 Introduction

This chapter has been introduced the study by outlining the background and motivation for cloud-based intrusion detection by defining the research question and objectives and showing the need for hybrid ensemble machine learning (ML) and explainable AI (XAI) to improve accuracy, performance and interpretability in modern cloud network security environments.

1.1 Background and Motivation

Modern cloud environment has become very dynamic and complex infrastructures whereby, huge quantities of logs and resource metrics are constantly produced (Vankayalapati, 2025) by virtual machines, containers, micro services and distributed applications (Mathur, 2024). With organisations becoming more dependent on cloud platforms as the means to scale and provide flexibility in operations, the risk of performance failure, configuration errors, cyber-attacks, and abnormal system behaviours are also increasing. The rule-based monitoring tools

are not efficient to analyse high dimensional cloud data or identify hidden anomalies because the patterns of cloud workloads evolve rapidly (De Gasperis and Facchini, 2025). The advances of machine learning techniques effort to tackle successive log dependencies and are not very transparent in their predictions. According to deep learning architectures, especially CNNs and LSTMs, there has been a demonstrated high ability to learn spatial and temporal patterns in complex system logs. But due to their black-box nature, they pose a problem to interpretability in security-sensitive settings. Such a gap has resulted in the growing demand to develop intelligent, adaptive and explainable systems of anomaly detection that can operate on a real-time basis in the contemporary cloud environments. The study has been motivated by the increasing complexity and scale of cloud networks which demand intelligent and scalable intrusion detection systems. It aims to increase security by combining hybrid ensemble learning with explainable AI for accurate and interpretable attack detection.

1.2 Research Question

How effectively can a hybrid ensemble machine learning–based network intrusion detection system which is deployed in a cloud environment, improve attack classification accuracy, reduce detection latency and enhance model interpretability on the UNSW-NB15 dataset as compared to individual classifiers?

1.3 Research Objectives

This study has few research objectives which includes:

1. To design and implement a cloud-based network intrusion detection system using a hybrid ensemble of ML classifiers for effective identification of normal and malicious network traffic.
2. To evaluate and compare the performance of individual ML models and the proposed hybrid ensemble approach in terms of metrics.
3. To enhance and interpret IDS decisions by using Explainable AI techniques (LIME) in order to improve transparency, trust, and usability of the model in real-world cloud security environments.

1.4 Structure of the Study

This section outlines the overall organization of the report by briefly describing the purpose and content of each chapter.

1. Chapter 1 – Introduction: This chapter introduces the research context, explains the motivation behind the study and defines the research questions and objectives.
2. Chapter 2 – Literature Review: This chapter examines previous studies related to network intrusion detection, cloud security, ML models and XAI techniques.

3. Chapter 3 – Methodology: This chapter shows the research framework, dataset selection, preprocessing methods, model design and explainability approach which has been adopted in the study.
4. Chapter 4 – Design Specification: This explains the system architecture, design decisions, cloud framework and justification of the proposed hybrid ensemble model.
5. Chapter 5 – Implementation: This explains the tools, technologies, model implementation, XAI integration and AWS-based deployment process.
6. Chapter 6 – Results and Analysis: This chapter presents experimental findings, performance evaluation and comparative analysis of the proposed and baseline models.
7. Chapter 7 – Conclusion and Future Scope: This chapter concludes the study by summarizing outcomes, discussing limitations and suggesting potential future works.

2 Related Work

This chapter reviews the evolution of cloud anomaly detection, explains the diverse nature of anomalies in dynamic cloud environments and shows the growing role of XAI in improving transparency and trust. It connects traditional limitations, modern ML-based detection needs and XAI-driven interpretability for secure, reliable cloud operations.

2.1 Cloud Anomaly Detection in Modern Computing

Detection of anomalies in clouds is an important part of modern computing with the further rise of the scale, complexity, and heterogeneity of a cloud environment (Lian et al., 2025). Cloud systems have become source of enormous volumes of operational metrics, logs and events in response to the rapid proliferation of virtualization, containerization and distributed architecture (Queiroz et al., 2024). These flow of data should be monitored in order to provide reliability of the system, stability in performance, and security (Hao et al., 2022). Traditional systems were mostly rule-based when it came to detecting anomalies and used fixed thresholds or manual signatures. Nevertheless, these approaches are no longer applicable in cloud ecosystems in which workloads change dynamically, resources are scaled automatically and usage patterns change significantly over time.

Recent cloud anomaly methods are concentrated on the detection of variations between normal behaviour in multivariate and high dimensional data (Suboh et al., 2023). It also involves the identification of unforeseen spikes, degradation, resource abuse, unauthorized operations as well as unnatural microservice interactions. It is a problem because cloud environments are extremely elastic and thus present ever-changing baselines which cannot be recorded using fixed monitoring tools. This has led to studies that are more adaptive and learning-based, which can comprehend patterns, correlations and temporal behaviours. These smart methods increase false alarm reduction, early detection and proactive decision making

in cloud operations. Problems with large-scale cloud infrastructures may be prevented by effective anomaly detection to maintain service availability.

2.2 Nature of Anomalies in Cloud Computing Environments

Anomalies in the behaviour of cloud computing environments become apparent when the behaviour of the system does not conform to what is expected, which is often an indication of operational problems, performance constraints, of possible security threats (Alloghani, 2024). Cloud systems are distributed in nature and comprise of virtual machines, containers, storage clusters, and interconnected services, which are functioning across multiple levels. Anomalies are very diverse in nature because each component creates logs and metrics representing different states of operations. Typically it can be performance anomalies when a CPU spike or memory leakage occurs, security anomalies when access patterns seem suspicious or unauthorized requests are made, and operational anomalies when the configuration varies or a service fails or the allocation of resources is disrupted.

Anomalies in cloud workloads are not consistent or predictable because the workload is dynamic (Gadhavi and Bhavsar, 2022). Due to the auto-scaling events, the changing user traffic and the varying requirements of resources, the behaviour of the system is constantly changing and it is hard to draw definite limits of the normal behaviour (Ogundipe, 2024). There are as well numerous anomalies that come into view in a gradual manner, in creating a slight deviation over time and not a sudden shift. Also, the existence of dependencies between microservices leads to cascading effects, where any anomaly in one component is reflected to other components. To investigate the nature of cloud anomalies, it is necessary to examine the spatial distributions of features as well as an analysis of how a sequence of events is related to time. It is this complexity that underscores the importance of advanced detection tools that have the ability to pick up the changing behaviours in real time.

2.3 Explainable AI (XAI) for Cloud Security and Model Transparency

In the recent developments of cloud, IoT, and autonomous system security, Explainable Artificial Intelligence (XAI) became one of the essential facilitators of enhancing anomaly detection, model transparency, and reliable decision-making. The overall ten-article body indicates a solid trend of introducing interpretability into AI-based security systems working in centralised and federated, IoT- Edge-Cloud, encrypted, vehicular, and fully autonomous settings. A similar study (Wewaldeni Pathirannehelage, 2022) emphasized on the nature of explanations in centralized versus federated learning through the use of SHAP to evaluate the anomaly detection models trained on the UNSW-NB15 and NSL-KDD datasets and found that the level of client-level data distribution can be a major factor in influencing interpretability. In study (Namrita Gummadi et al., 2024), a wide XAI-based anomaly detection framework of IoT systems was proposed based on both single and ensemble AI models and seven interpretability techniques that showed better understanding of the features in both manufacturing and botnet datasets. In addition to this, the study (Roshan and Zafar, 2022) applied Kernel SHAP to maximize network anomaly detection on CICIDS2017 and demonstrated that explainability can directly positively impact the accuracy and overall

performance of the model. To overcome the issue of encrypted traffic observation, research (Singh et al., 2025) suggested a model-agnostic ML framework, which adds SHAP to explain prediction on the three encrypted datasets, and demonstrated that it is possible to detect anomalies transparently even without inspecting the payloads. In more comprehensive reviews such as study (Chukwuemeka Nwachukwu et al., 2024), the relevance of AI-based anomaly detection in cloud systems has been highlighted to grow, with the identified issues including computational cost, scalability, and heterogeneity of data, and the suggested future solutions being the incorporation of XAI, blockchain, and self-supervised learning as shown in Table 1. Study (Nazat et al., 2024) generalized interpretability to Autonomous Vehicle (AV) networks by describing an XAI-ADS system that united SHAP and LIME with new feature selection methods with better interpretability and model efficiency on several AV datasets. In study (Nguyen et al., 2024), Montimage Monitoring Tool (MMT) was augmented with SHAP, LIME, and SHAPASH to give useful explanations of network anomalies in a bid to fill the transparency gaps in live traffic monitoring. Study (Baimukhanov et al., 2025) also extended the use of XAI by suggesting a multi-level IoT-Edge-Cloud anomaly detection system using Isolation Forest, LOF, and interpretable visual analytics via smart web interface that was tested using an IoT-healthcare case study with AUROC 1.00. Finally, the research (Nazat, 2024) included one of the broadest XAI security work in the context of AV networks, combining SHAP-based feature selection, XAI method benchmarking, LSTM- based platoon security, and DAG-based resilience assessment, all of which collectively showed that explainability enhanced defense against the different network structures. Collectively, these studies show that XAI is necessary to fill the gap between high-performance ML models and the transparency demanded in sensitive cloud and networked systems. Despite the persistent issues of computational load, data size, and real-time performance, the aggregation of evidence demonstrates clearly that XAI is more likely to improve the trust, interpretability, and detection accuracy and resilience in the contemporary security ecosystems in the cloud.

Table 1: Summary Table

Study	Purpose	Proposed Approach	Challenges Addressed	Key Results
(Wewaldeni Pathirannehelage, 2022)	To analyze how XAI explanations differ between centralized and federated learning models for anomaly detection.	Used Deep Neural Networks as black-box models with SHAP explainability; evaluated UNSW-NB15 and NSL-KDD under different FL client anomaly ratios.	Data scarcity in FL, inconsistent explanations across distributed clients, accuracy variations.	Showed that FL explanations significantly change with anomaly proportions; demonstrated XAI utility in both FL and centralized settings.
(Namrita Gummadi et al., 2024)	To detect and interpret anomalies in IoT systems using AI and XAI techniques.	Used single and ensemble AI models; applied seven XAI methods for feature importance; evaluated on manufacturing and IoT-botnet datasets.	Difficulty identifying anomalies in heterogeneous IoT data; need for interpretable feature insights.	Accurate anomaly detection with top features identified across datasets; released code for community use.
(Roshan and Zafar, 2022)	To detect and explain network	Trained two models (Model_1 &	Understanding anomaly reasoning;	OPT_Model achieved 0.90

	anomalies using KernelSHAP and improve model performance.	OPT_Model) on CICIDS2017; applied KernelSHAP to interpret and optimize model behavior.	improving unsupervised performance metrics.	accuracy and 0.76 F-score in unsupervised mode; SHAP helped refine model.
(Singh et al., 2025)	To create an interpretable ML framework for detecting anomalies in encrypted network traffic.	Used multiple ML classifiers with SHAP for post-hoc explanations across three encrypted datasets.	Encryption prevents payload inspection; need for transparent ML decisions.	SHAP revealed key influential features, improving trustworthiness; strong accuracy across classifiers.
(Chukwuemeka Nwachukwu et al., 2024)	To review AI-based anomaly detection methods in cloud environments.	Surveyed ML, DL, hybrid models; discussed federated learning and model optimization as solutions.	Scalability, data quality issues, dynamic cloud behaviors.	Highlighted AI's strength in detecting complex anomalies and outlined future directions (XAI, blockchain, IoT).
(Nazat et al., 2024)	To build an XAI-enhanced anomaly detection framework for AV networking security.	Applied SHAP- and accuracy-based feature selection; evaluated six black-box models; used global and local XAI.	Lack of interpretability in AV anomaly systems; growing cyber risks.	Proposed feature selection methods outperformed six existing techniques; efficient and interpretable models achieved.
(Nguyen et al., 2024)	To enhance Montimage Monitoring Tool (MMT) with XAI-based interpretability for network anomaly detection.	Integrated SHAP, LIME & SHAPASH into MMT; conducted initial experiments for explanation quality.	Real-time detection in massive network data; lack of transparency in AI classifications.	Provided clear feature contributions and interpretable results; improved analyst trust.
(Baimukhanov et al., 2025)	To improve anomaly detection across IoT–Edge–Cloud systems with XAI and layered processing.	Multi-tier design: IoT microcontrollers → Edge preprocessing → Cloud ML (Isolation Forest, LOF) → SPA dashboard; XAI for interpretability.	High data volume, latency reduction, sensor reliability, secure multi-layer interactions.	Achieved AUROC 1.00 (Isolation Forest); reduced latency via edge processing; improved transparency using XAI.
(Nazat, 2024)	To develop integrative XAI-based and cybersecurity frameworks for AV and VANET ecosystems.	SHAP-based feature selection; SHAP vs LIME benchmarking; LSTM-RSU platoon security system; DAG-based AV network attack analysis.	Lack of interpretability, real-time anomaly detection, secure platoon coordination, DAG network vulnerabilities.	Novel feature selection outperformed SOTA; robust platoon defense against MITM/DoS; DAG resilience insights provided.

2.4 Synthesis and Research Gaps

The literature reviewed shows a high level of advancement in the field of cloud anomaly detection and the combination of Explainable AI, but there are still some significant gaps in the number of researches. The current literature is based on single or a few ensemble models which are not always able to make inferences across varying and overlapping attacks that are prevalent in changing cloud environments. Despite the research on ensemble and hybrid methods, few studies have provided a systematic combination of many tree-based learners with an effective meta-classifier to achieve a compromise on accuracy, robustness and scalability towards cloud-based intrusion detection. Also, most of the works focus on the performance of the detectors but fail to adequately examine real-life issues of deployment in cloud services like latency, throughput, and cloud resources scalability, which are crucial to service clouds. This study fills these gaps by proposing a hybrid ensemble system of intrusion detection which is cloud-deployable, that uses SMOTE based balancing and explainability with LIME which results in a better system that is more practical, interpretable and relevant to modern cloud security contexts.

3 Methodology

The chapter explains research technique that will be used in the study, which incorporates CRISP-DM research design framework, data description and acquisition, data preprocessing, data visualization, class imbalance with SMOTE, and data splitting processes to facilitate accurate, scalable, and reproducible experiments of cloud-based intrusion detection.

3.1 Research Design Framework

This research was primarily aimed at coming up with and designing NIDS that is capable of detecting the contemporary cyber-attacks within cloud-based network settings with high precision. The Cross Industry Standard Process for Data Mining (CRISP-DM) has been used as the structure and systematic framework of the research design and consists of six steps; Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment.

The methodology has concentrated on three key areas that involve: high detection rate on various classes of attacks, capability to manage class imbalance and low-latency capability that can be used in the cloud mode. It is represented by the UNSW-NB15 dataset containing realistic network traffic and modern attacks behavior that makes it practically relevant.

This system has been developed based on Python-based ML frameworks and tested on a cloud-compatible platform with AWS infrastructure. To enhance the external validity and reproducibility the methodology is structured in a way that it can be completely reproduced on publicly available datasets, libraries open to the open source as well as to the virtual cloud. This strategy ensures that the suggested NIDS can be validated on its own at a low cost.

3.2 Dataset Description and Data Acquisition

This research relied on the UNSW-NB15 dataset[1] to assess the efficiency of the proposed Network Intrusion Detection System because it can give a realistic insight into both the modern network traffic and recent cyber-attack scenarios. The data was created inside the Cyber Range Lab within the Australian Centre of Cyber Security (ACCS) as the IXIA PerfectStorm tool that generates a composite blend of actual normal network traffic and the synthetic generation of attack behaviors.

This was recorded via Tcpdump network capture tool which generated about 100 GB of raw packet capture (PCAP) files. These raw packets were then analyzed with the help of Argus and Bro-IDS tools, and the twelve feature extraction algorithms were used to produce a total of 49 network traffic features with respective class labels. There are nine types of attacks, such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms in the dataset, and normal traffic cases.

The total amount of data is 2,540,044 records that were split into four CSV files, namely: UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, and UNSW-NB15_4.csv. The supplementary supporting files are UNSW-NB15_GT.csv that has the ground truth labels and UNSW-NB15-LIST-EVENTS.csv that has information about the events. To achieve consistency in the experiments, the predefined training and testing partitions were established and included 175,341 and 82,332 records respectively.

3.3 Data Preprocessing

Preprocessing of the raw UNSW-NB15 data was conducted to convert it to a formatted and machine-learn format. First, non-informative and irrelevant attributes like id, service and label were discarded off the dataset as these attributes do not add any value to the prediction learning process, and they can bring bias or redundancy to the dataset. This was followed by examination of the data in order to determine categorical characteristics. Label Encoding was used to extract all non-numerical attributes and put them in numerical form (Luo et al., 2020). This needed to be done since machine learning models cannot perform calculations without having numerical data. Application of the LabelEncoder was done on a column-to-column basis in order to guarantee uniform and effective conversion of categorical values to integer equivalents with the preservation of relational data. The dataset following encoding comprised 42 pure numerical features (traffic-based, flow-based and content-based features) such as duration (dur), protocol (proto), packet statistics (spkts, dpkts), byte counts (sbytes, dbytes), rate features and connection tracking measures. Attack variable attack_cat indicates the various types of attacks, thus allowing multiclass intrusion detection.

[1] https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15

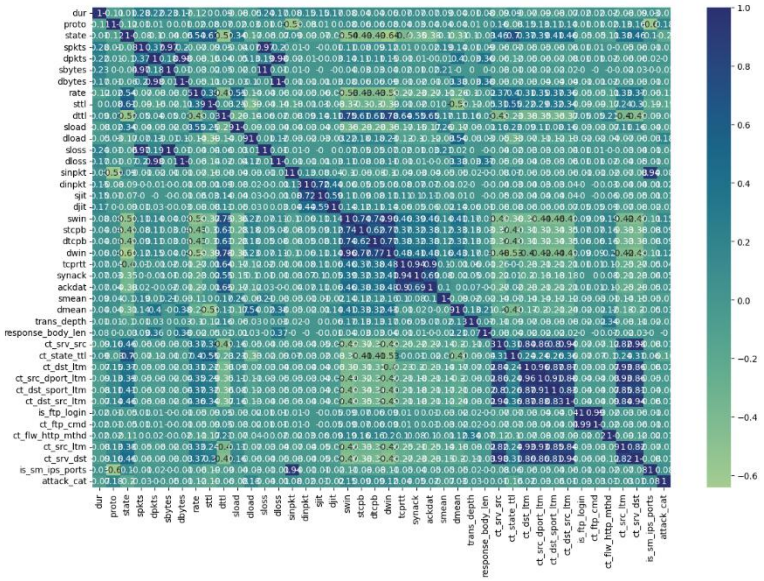

Figure 1: Correlation Heatmap of Network Traffic Features

Figure 1 shows a heatmap of correlation, which represents graphically the pairwise correlation of all the numerical features of the UNSW-NB15 dataset. This value is utilized to examine the strength and the direction of associations between attributes of network traffic with the intensity of the colors used being used to denote the extent of correlation. The heatmap can also be used to locate many correlated features that can lead to redundancy and multicollinearity thus adversely affecting the model performance. It also gives a justification of the feature selection and dimensionality reduction as it accentuates unimportant or poorly related features.

3.4 Data Visualization

Figure 2 is a bar graph that shows the distribution of the target classes labels used in the intrusion detection dataset with label 0 being normal traffic and label 1 is attack traffic. This number is applied to study the balance of classes in the data before training the model.

Figure 3 illustrates a bar chart of the frequency of the most frequently utilized network protocols in the UNSW-NB15 dataset. The figure is applied to learn protocol-level traffic behavior, which TCP and UDP control the total network communication. It is significant to analyze the protocol distribution as the various types of attacks are usually based on a particular protocol. The obvious asymmetry of protocols is the reason protocol-based feature analysis is justified and protocol-based intrusion models understand the realistic traffic behavior.

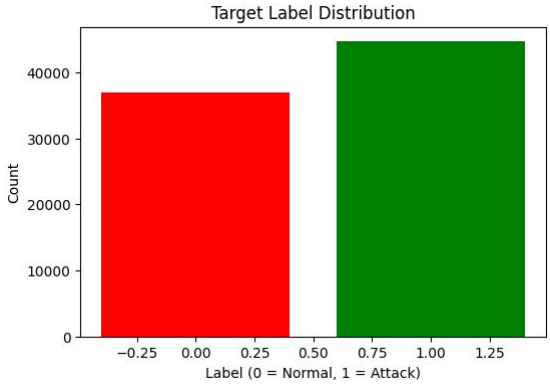

Figure 2: Target Label Distribution

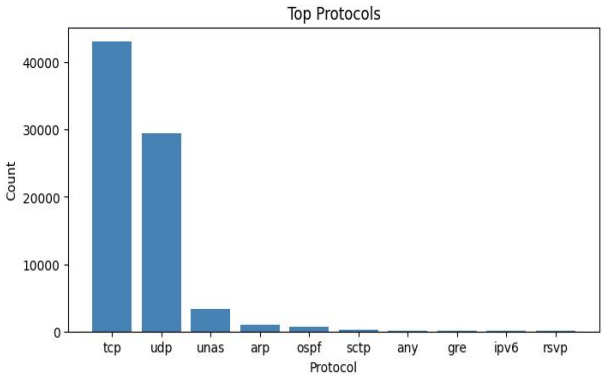

Figure 3: Top Network Protocols

In Figure 4, a bar chart is used to show the distribution of various types of attacks and normal traffic in the UNSW-NB15 dataset. This parameter is employed to examine the occurrence of the types of attacks, which are: Generic, Exploits, Fuzzers, and DoS against normal network behavior. This imbalanced distribution underscores the presence of class imbalance in most categories of attack, and therefore may impact on the performance of multi-class classifications.

Figure 5 plots a bar chart of the distribution of the network connection states in the UNSW-NB15 data. The number is applied to the session-level behavior analysis by demonstrating the frequency of occurrence of various TCP states like FIN, INT, and CON in the network traffic. The pre-eminence of FIN and INT states means there will be a high number of completed or interrupted connections, which is common in the real network environment.

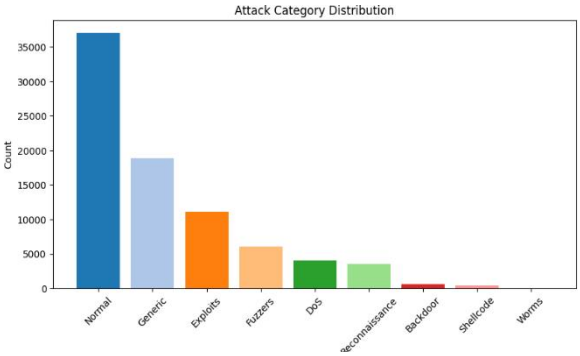

Figure 4: Attack Category Distribution

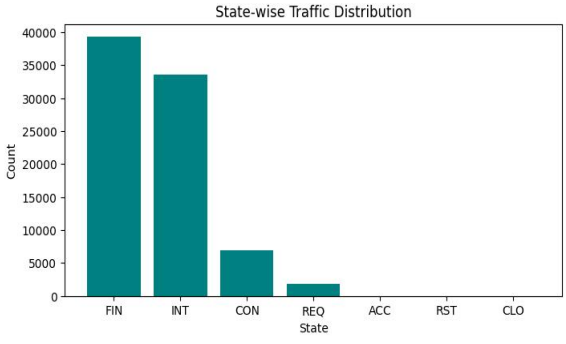

Figure 5: State-wise Network Traffic Distribution

3.5 Data Balancing and Data Splitting

To solve the challenge of class imbalance which existed in UNSW-NB15 dataset, Synthetic Minority Over-sampling Technique (SMOTE) was utilized in the data preprocessing phase (Sayegh et al., 2024) as shown in Figure 6. As shown in the figure, few categories of attacks were underrepresented as compared to majority classes, which may bias the learning process and cause less detection of minority attacks. SMOTE is used to create synthetic examples of minority classes by interpolating existing examples, therefore creating a more balanced

distribution of classes without data duplication. This method enhances better generalization of the model as well as equitable learning of all types of attacks. The dataset was split at the ratio of 80:20 to result in training and test subsets after the balancing. The train test split procedure was implemented with the shuffling option on and a random state that is fixed in order to be reproducible. This division approach will provide enough data to learn the models, and an independent test set will be available to provide fair performance assessment.

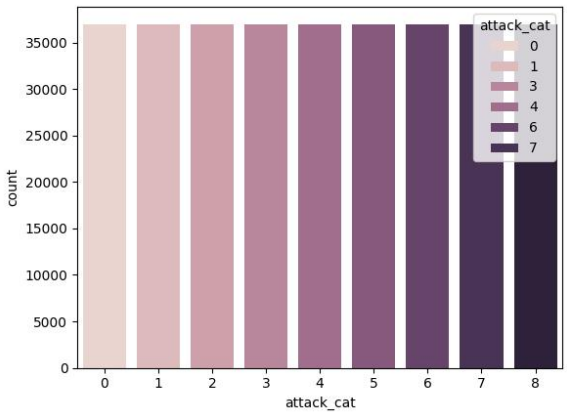

Figure 6: Class Distribution After SMOTE Oversampling

4 Design Specification

The chapter outlines the system architecture of the proposed cloud-based intrusion detection framework and reasons behind the chosen hybrid ensemble approach, cloud-focused implementation, data balancing approach, and the XAI integration to enable scalability, accuracy, robustness, and interpretability of modern cloud network security settings.

4.1 System Architecture and Framework

The proposed cloud-based NIDS on the AWS platform will have its end-to-end workflow as shown in Figure 7. The architecture starts with the UNSW-NB15 dataset, which is the realistic network traffic with normal and malicious activities. This data is firstly fed into the data loading component and then it is sent to data preprocessing layer. Some of the key functions that this layer carries out are data cleaning, feature encoding, normalization, and preparation in order to make the data machine learning model friendly. This processed information is then passed on to the model training module where the individual machine learning classifiers and the hybrid ensemble model are trained using the ready dataset. When the models are trained, it is then implemented in the Prediction layer to identify incoming traffic in case they are normal or attack instances. To solve the interpretability problem of ensemble models, an XAI is also added, which can give local and feature-wise explanations of specific predictions, which improves transparency and trust among analysts. After prediction and explanation, The entire workflow is created and managed with the help of AWS Cloud9, which is then run with the help of Amazon EC2 to get scalable computation and with the help of Amazon S3 to maintain the persistent stores of the datasets, results and

visual outputs. In general, this architecture shows an interpretable, scalable and cloud ready intrusion detection system.

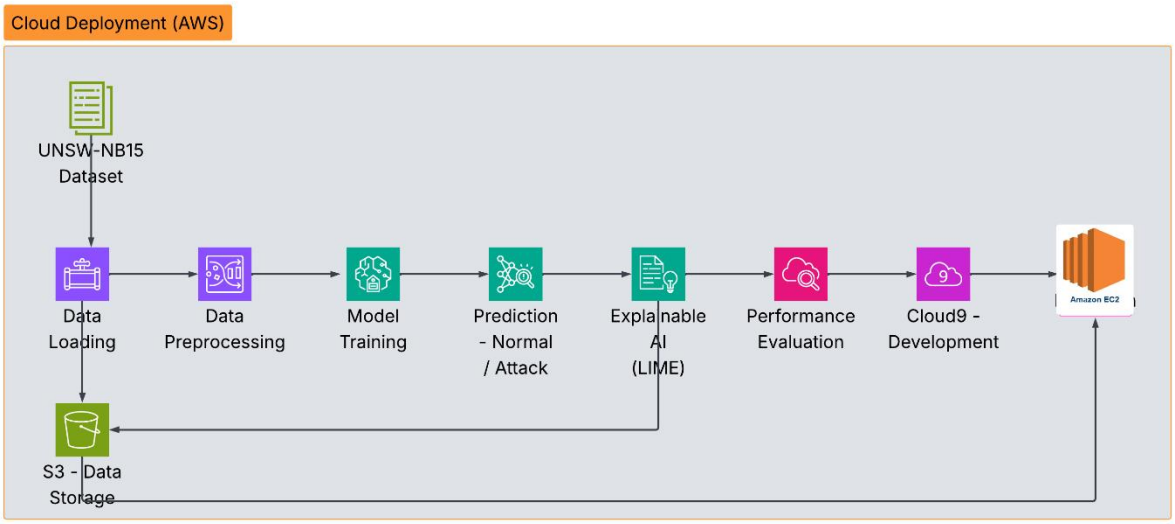

Figure 7: System Architecture Diagram

4.2 Justification for the Chosen Methodology

The presented methodology has been designed to solve the challenging task of the security of modern cloud-based networks, where the traffic patterns are extremely dynamic, heterogeneous, and large-scale. The traditional single classifier methods are in many cases ineffective in such settings because of their little capacity to model various attack behaviours and non-linear interaction of features. Thus, an ensemble learning model was selected, but the hybrid model was adopted to use the benefits of several classifiers that are complementary. The proposed hybrid model with XGBoost as a meta-classifier combined with Gradient Boosting, Random Forest, and AdaBoost as base learners lower both bias and variance and consequently, the suggested data model enhances detection accuracy and resistance to overlap attacks and evolving patterns of attacks that are typical in cloud traffic.

Cloud services produce large amounts, distributed and time-span network information that needs a scalable, computationally-efficient methodology. The cloud-centric design is justified by the fact that the AWS resources like EC2, Cloud9, S3 allow allocating its resources elastically, processing in parallel, and storing big datasets reliably (Kodi and Anumolu, 2025). This can be assured that the proposed intrusion detection system can be able to work in the actual cloud infrastructure as opposed to being an offline solution.

Also, intrusion datasets used to address class imbalance was considered with SMOTE to enhance the minority attack detection, which is essential in cloud security. The fact that the methodology also includes Explainable AI with the use of LIME further supports the methodology and makes it more transparent and trustworthy, allowing security analysts to comprehend and confirm intrusion predictions in cloud systems.

5 Implementation

This chapter describes the deployment of the proposed system of intrusion detection, including tools, technologies, and cloud environment details, the implementation of individual and hybrid ensemble machine learning models, the implementation of the system in the XAI with the usage of LIME and the implementation of the system by means of the AWS services that are reliable and can be easily scaled.

5.1 Tools, Technologies, and Environment

The proposed study has implemented using a Python-based ML environment to have flexibility, scalability and reproducibility of experiments as shown in Table 2. Warning suppression and time analysis utilities have included to maintain clean execution and performance tracking. This environment provides a strong, powerful and research-oriented framework suitable for large-scale network intrusion detection experiments.

Table 2: Tools and Technologies Used

Category	Tool / Technology	Purpose
Programming Language	Python (< 3.6.3)	Core implementation environment
Data Processing	Pandas, NumPy	Data handling and numerical computation
Visualization	Matplotlib, Seaborn	Exploratory data analysis
Machine Learning	Scikit-learn	Model training, evaluation, preprocessing
Ensemble Learning	XGBoost, MLXtend	Boosting and stacking classifiers
Data Balancing	Imbalanced-learn (SMOTE)	Handling class imbalance
Feature Selection	Mutual Information, SelectPercentile	Relevant feature extraction
Utilities	Pickle, Time, Warnings	Model persistence and execution control

5.2 Implementation of ML Models

5.2.1 Random Forest

The classifier of random forest was adopted as a baseline method of ensemble learning based on its robustness and the capability to deal with high-dimensional network traffic information (Akhtar et al., 2023). Training of the model was done with few decision trees, with each tree being built on a random sample of features and samples. It is due to this randomness that overfitting is minimized, and generalization is enhanced. Random Forest is best applicable in intrusion detection applications since it is able to learn non-linear correlation between network characteristics. For the implementation the model was trained using the balanced training data and tested on unknown test data to make predictions of attack types. It also has

an implicit feature importance because of its tree-based structure and is therefore useful in the understanding of network attributes that are most useful in intrusion detection. Random Forest is therefore a good and interpretable model to use in the first performance comparison in this study.

5.2.2 AdaBoost Classifier

To improve the performance of the AdaBoost classifier, the implementation of the AdaBoost classifier was aimed at prioritizing the hard-to-classify individuals. AdaBoost is a sequential combination of numerous weak learners, with each successive learner laying more weight on the previously wrongly classified samples. The adaptive learning mechanism has been found to be especially useful in intrusion detection cases where the minority attack patterns would be neglected. AdaBoost trained on the SMOTE-balanced data was used in this analysis to enhance sensitivity to the rare types of attacks. The weights of samples are adjusted through repeated adjustments to create the model which learns intricate decision boundaries which more accurately differentiate normal and malicious traffic.

5.2.3 XGBoost Classifier

XGBoost was deployed as an effective gradient boosting model reputed to be highly efficient and accurate in its predictions (Noorunnahar et al., 2023). The model constructs the decision trees in a sequential fashion as it minimizes a regularized loss which assists in regulating overfitting. In the present work, the XGBoost was used to learn complicated relationships between the features of network traffic like protocol type, flow duration, and packets statistics. It is very well adapted to intrusion detection work due to its capacity to deal with scanty and imbalanced data. Also, XGBoost is as well associated with parallel processing and efficient tree building which makes it easier and faster to train and also it has higher scales. The balanced training set was used to train the model and the test set was used to test the classification ability of the model. XGBoost was chosen as one of the models with the highest performance across security-related datasets, which will result in the high detection accuracy and strong performance.

5.2.4 Gradient Boosting Classifier

The Gradient Boosting Classifier was used in a series of rounds of enhancement of the performance of prediction by reducing the number of classification errors through gradient descent optimization. Gradient Boosting uses trees built successively as opposed to Random Forest, which builds trees independently, with every successive tree correcting the errors of the previous ensemble. This renders it useful in grabbing delicate attack pattern in network traffic. In this work, the model has been trained with few estimators to compromise between accuracy and cost of computation. Gradient Boosting is a more appropriate technique in intrusion detection since non-linear and complex relationships between variables, including traffic rates and connection tracking measures, can be modelled. Its presence gives it a comparative study on other boosting methods and assists in measuring the effectiveness of sequential methods of learning to identify various cyberattacks.

5.2.5 Hybrid Ensemble Model

Hybrid Ensemble Model was applied in a stacking-based method and was used to benefit the advantage of several classifiers (Chaurasia and Pal, 2021). Gradient Boosting, Random Forest and AdaBoost were utilized as base learners and XGBoost as the meta-classifier was deployed in the study. The base models have independently learnt varied patterns during the training data which represented varied aspects of network behavior. Their forecasts were subsequently aggregated and submitted to the meta classifier which trained to know how best to combine such outputs to come up with the final decision. The architecture improves the detection accuracy by decreasing the bias and variance of separate models.

5.3 Explainable AI Integration

To make the suggested intrusion detection system more transparent and interpretable, Explainable Artificial Intelligence (XAI) was deployed with the help of the LIME framework. LIME has been used to explain single predictions made by the stacking ensemble classifier. A LimeTabularExplainer was launched with the training data and the names of the features matched the names of the input features, and the names of the classes matched the multi-class categories of attack. Continuous variables were categorized in order to enhance clarity of explanation.

One case of testing data was taken and its probability of being predicted by the stacking model which was trained was taken. LIME produced a local explanation approximating the action of the complex model around the chosen instance with a linear model that can be interpreted. The description presents the top ten features that have had the greatest contribution towards the predicted class. Class 0 which was predicted to be the output of the chosen instance represented normal traffic.

The contribution scores indicate the extent to which individual features like ct_dst_sport-ltm, ct-src-dport-ltm, proto and swin have affected the decision of classification either negatively or positively. These descriptions were put in the tabular format and this was saved in the form of a CSV file to document and further analyze it. It enhances the credibility of a model by offering to a human knowledge to explain predictive behavior.

Figure 8: LIME-Based Local Explanation of Stacking Model Prediction

Figure 8 presents a LIME visualization plot which displays prediction probabilities for different classes by showing that the instance is classified as Class 0 with the highest confidence. It further shows the most influential features contributing toward and against the

predicted class along with their corresponding values. This figure shows how LIME has been implemented to interpret complex ensemble models by approximating local decision boundaries.

5.4 AWS Deployment

5.4.1 Amazon Cloud9

The main development environment on which the intrusion detection system was implemented was AWS Cloud9. It offers a browser-based IDE, which allows local system-independent code, debug, and execution of Python-based machine learning workflows. Cloud9 is closely bound up to the services of AWS that enables easy access to EC2 and S3 resources. This environment makes it easy to collaborate, allows runtime configuration to be consistent and makes experimentation quick. With Cloud9, local hardware constraints are removed and the entire model pipeline consisting of preprocessing, training, and explainability can be developed and tested in the cloud infrastructure.

5.4.2 Amazon EC2

The EC2 of Amazon was also used to offer scalable computing capabilities necessary to process the UNSW-NB15 data and to train ensemble machine learning models. An EC2 instance has been configured with adequate CPU and memory to effectively carry out data preprocessing, SMOTE oversampling, model training and evaluation processes. EC2 provides elasticity of resource allocation and employs high performance computing in large scale datasets. It is cost-effective as it is on demand but remains reliable. Implementation of the intrusion detection system on EC2 indicates that security analytics workloads can be executed in the real cloud setting.

5.4.3 Amazon S3

Amazon S3 was used as the central storage of dataset, results of the experiment and visualizations generated. The UNSW-NB15 dataset, along with confusion matrices, distribution plots and evaluation results have been securely stored in S3 buckets. S3 is very durable, scalable, and can easily access high volumes of data hence useful when it comes to machine learning pipelines. The system places a space between storage and computation, which results in improved data management and fault tolerance. This implementation promotes the effective retrieval of data and durability of the findings of research.

6 Evaluation

In this chapter, the proposed intrusion detection system is evaluated on individual and hybrid ensemble models on a basis of classification measures, latency and throughput analysis, and confusion matrices, which show that the models are effective, robust, and performance trade-offs in multi-class network intrusion detection.

6.1 Experiment 1: Classification Performance Evaluation

This experiment has been evaluated the classification performance of the implemented ML models using standard multi-class evaluation metrics. These metrics has been provided a balanced assessment by equally considering all attack classes by making them suitable for imbalanced intrusion detection datasets. As shown in the results in Table 3 the Hybrid Ensemble Classifier has achieved the highest performance across all metrics, with an accuracy and macro F1-score of 0.90 by showing its strong ability to capture complex attack patterns by combining multiple learners. Random Forest also performs well showing strong generalization with an accuracy of 0.87. XGBoost showed consistent and stable results across all metrics while Gradient Boosting has been achieved moderate performance.

Table 3: Classification Performance Comparison of ML Models

Model	Accuracy	Macro Precision	Macro Recall	Macro F1
Random Forest	0.87	0.88	0.87	0.87
AdaBoost Classifier	0.62	0.63	0.62	0.6
XGBoost Classifier	0.82	0.82	0.82	0.82
Gradient Boosting Classifier	0.74	0.75	0.74	0.74
Hybrid Ensemble Classifier	**0.9**	**0.9**	**0.9**	**0.9**

6.2 Experiment 2: Latency and Throughput

This experiment evaluates the performance of different ML models in terms of latency per prediction and throughput which are important metrics for real-time intrusion detection systems. Latency measures the time taken by a model to generate a single prediction while throughput shows the number of predictions processed per second. As shown in Table 4, XGBoost and Random Forest has been achieved the lowest latency values by making them suitable for time-sensitive environments. Random Forest also shows the highest throughput by showing powerful parallel processing. Whereas AdaBoost and the Hybrid Ensemble model showed higher latency and lower throughput due to increased model complexity and sequential learning type of behaviour. Although the Hybrid Ensemble showed reduced speed it balances performance with improved detection capability by making it suitable where accuracy is prioritized over real-time constraints.

Table 4: Latency and Throughput Performance of ML Models

Model	**Latency per Prediction (ms)**	**Throughput (Predictions/sec)**
Random Forest	0.9371	837,319.43
AdaBoost Classifier	20.6522	48,141.22
XGBoost Classifier	0.8159	521,985.70

Gradient Boosting Classifier	1.2476	356,424.14
Hybrid Ensemble Classifier	23.4709	10,604.58

6.3 Experiment 3: Confusion Matrices

Figure 9 shows the confusion matrix of the Random Forest model, which shows accurate and inaccurate predictions by the model based on classes. The high values of the diagonal show that there has been a high level of classification in most classes. Minor misclassifications take place between the similar types of attacks, which is a reflection of the overlapping traffic patterns. All in all, the performance is strong with taken care of classification errors in intrusion in complex multi-class problems.

The confusion matrix of AdaBoost classifier as shown in Figure 10 depicts a greater rate of misclassification with several categories of attacks than the random forest. Large off-diagonal values imply inability to distinguish between complex and overlapping patterns of attacks. This is indicative of the sensitivity of AdaBoost to noisy and imbalanced data, which leads to the decrease in the overall classification performance.

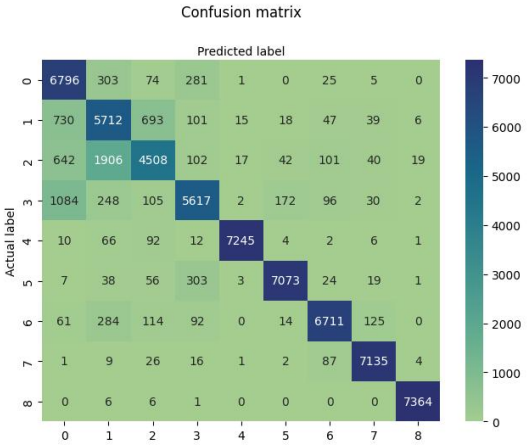

Figure 9: Confusion Matrix

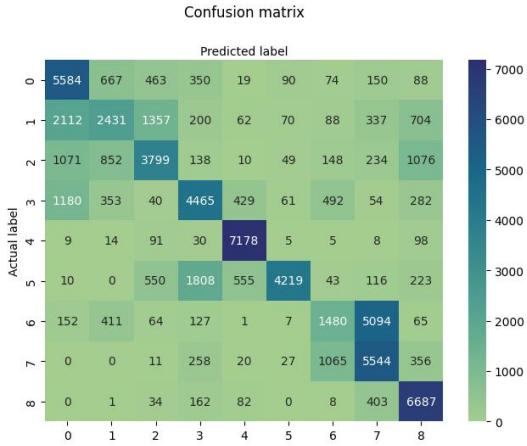

Figure 10: Confusion Matrix

The confusion matrix of XGBoost classifier as in Figure 11 shows that diagonal domination is high, and the ability to distinguish classes is better than in AdaBoost. Less misclassification in the majority of type categories of attacks, points to the efficient learning of more complicated patterns. Small variations exist between the attacks that are closely related, which is a manifestation of inherently similar traffic.

The confusion matrix of the Gradient Boosting classifier is shown in Figure 12 with an average level of diagonal dominance and some significant misclassifications between specific attack classes. Although the model can capture general intrusion trends, similar attack behavior overlaps bring complexity to classification and lead to greater error rates than XGBoost and Random Forest.

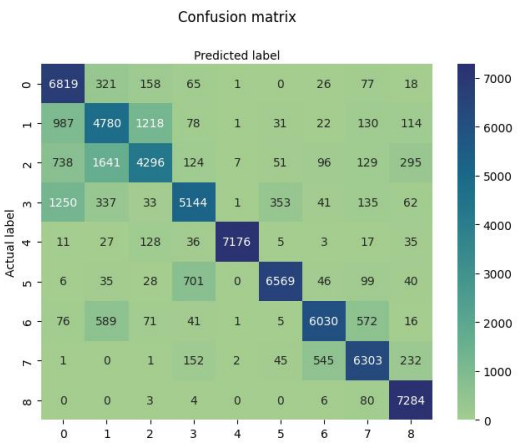

Figure 11: Confusion Matrix

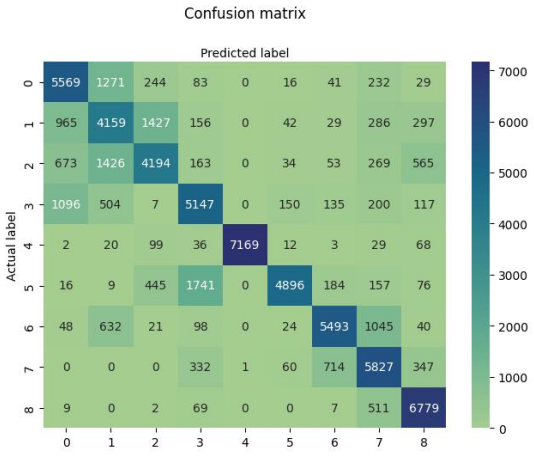

Figure 12: Confusion Matrix

The confusion matrix of the Hybrid Ensemble classifier in Figure 13 shows high levels of diagonal dominance and less misclassification among the attack categories. The lower off-diagonal errors reflect the good management of overlapping and complicated intrusion guidelines. This proves to be better robust and balanced than individual classifiers.

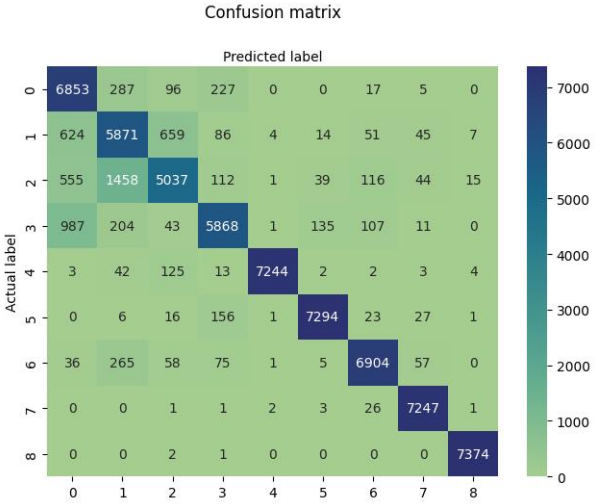

Figure 13: Confusion Matrix

7 Conclusion and Future Works

This study has introduced a cloud network intrusion detection system based on the hybrid ensemble machine learning model tested on the UNSW-NB15 dataset. The experimental findings prove that the ensemble model is more successful than the individual classifiers in terms of classification accuracy, attack detection capability, and general strength and thus, answers the research question. Integration of Gradient Boosting, Random Forest, and AdaBoost with XGBoost meta-classifier made it possible to achieve better generalization of a wide variety of attack types with acceptable latency and throughput in a cloud environment. In addition, XAI by using LIME improved the transparency of the model by generating

meaningful feature-level explanations of intrusion predictions, which contributes to a viable trust and deployment preparedness.

The study has some limitations in spite of these promising findings. The experiments were carried out on a benchmark dataset that was offline and real-time network traffic was not taken into account. Also, the deployment of the ensemble model to large-scale high-speed networks might encounter computational overhead, scalability issues. Future research will involve real time intrusion detection over live traffic streams, optimization of the model to work on edge and resource constrained models and incorporation of deep learning models like LSTM and attention based models. More research on the methods of advanced explainability and adaptive learning processes can also improve the performance related to detection of the changing cyber threats.

References

Akhtar, M.A., Qadri, S.M.O., Siddiqui, M.A., Mustafa, S.M.N., Javaid, S., Ali, S.A., 2023. Robust genetic machine learning ensemble model for intrusion detection in network traffic. Sci. Rep. 13, 17227. https://doi.org/10.1038/s41598-023-43816-1

Alloghani, M.A., 2024. Anomaly Detection of Energy Consumption in Cloud Computing and Buildings Using Artificial Intelligence as a Tool of Sustainability: A Systematic Review of Current Trends, Applications, and Challenges, in: Artificial Intelligence and Sustainability, Signals and Communication Technology. Springer Nature Switzerland, Cham, pp. 177–210. https://doi.org/10.1007/978-3-031-45214-7_9

Baimukhanov, S., Ali, H., Yazici, A., 2025. Enhancing ML-based anomaly detection in data management for security through integration of IoT, cloud, and edge computing. Expert Syst. Appl. 293, 128700. https://doi.org/10.1016/j.eswa.2025.128700

Chaurasia, V., Pal, S., 2021. Stacking-Based Ensemble Framework and Feature Selection Technique for the Detection of Breast Cancer. SN Comput. Sci. 2, 67. https://doi.org/10.1007/s42979-021-00465-3

Chukwuemeka Nwachukwu, Kehinde Durodola-Tunde, Chukwuebuka Akwiwu-Uzoma, 2024. AI-driven anomaly detection in cloud computing environments. Int. J. Sci. Res. Arch. 13, 692–710. https://doi.org/10.30574/ijsra.2024.13.2.2184

De Gasperis, G., Facchini, S.D., 2025. A Comparative Study of Rule-Based and Data-Driven Approaches in Industrial Monitoring. https://doi.org/10.48550/ARXIV.2509.15848

Gadhavi, L.J., Bhavsar, M.D., 2022. Adaptive cloud resource management through workload prediction. Energy Syst. 13, 601–623. https://doi.org/10.1007/s12667-019-00368-6

Hao, Q., Zhang, Z., Xu, D., Wang, J., Liu, J., Zhang, J., Ma, J., Wang, X., 2022. A Hardware Security-Monitoring Architecture Based on Data Integrity and Control Flow Integrity for Embedded Systems. Appl. Sci. 12, 7750. https://doi.org/10.3390/app12157750

Kodi, D., Anumolu, V.R., 2025. Leveraging AWS Cloud-Based Ecosystems to Drive Sustainable Innovation:, in: Sorayyaei Azar, A., Kant Gupta, S., Taherdoost, H.,

Alhamaty, F. (Eds.), Advances in Computational Intelligence and Robotics. IGI Global Scientific Publishing, pp. 353–372. https://doi.org/10.4018/979-8-3373-3952-8.ch015

Lian, L., Li, Y., Han, S., Meng, R., Wang, S., Wang, M., 2025. Artificial Intelligence-Based Multiscale Temporal Modeling for Anomaly Detection in Cloud Services. https://doi.org/10.48550/ARXIV.2508.14503

Luo, S., Miao, D., Zhang, Z., Wei, Z., 2020. Non-numerical nearest neighbor classifiers with value-object hierarchical embedding. Expert Syst. Appl. 150, 113206. https://doi.org/10.1016/j.eswa.2020.113206

Mathur, P., 2024. Cloud Computing Infrastructure, Platforms, and Software for Scientific Research, in: Ahmad, K.A., Hamid, N.A.W.A., Jawaid, M., Khan, T., Singh, B. (Eds.), High Performance Computing in Biomimetics, Series in BioEngineering. Springer Nature Singapore, Singapore, pp. 89–127. https://doi.org/10.1007/978-981-97-1017-1_4

Namrita Gummadi, A., Napier, J.C., Abdallah, M., 2024. XAI-IoT: An Explainable AI Framework for Enhancing Anomaly Detection in IoT Systems. IEEE Access 12, 71024–71054. https://doi.org/10.1109/ACCESS.2024.3402446

Nazat, S., 2024. ANOMALY DETECTION AND EXPLAINABLE AI FOR ENHANCED SECURITY IN AUTONOMOUS VEHICLE NETWORKS 7284910 Bytes. https://doi.org/10.25394/PGS.27960993.V1

Nazat, S., Li, L., Abdallah, M., 2024. XAI-ADS: An Explainable Artificial Intelligence Framework for Enhancing Anomaly Detection in Autonomous Driving Systems. IEEE Access 12, 48583–48607. https://doi.org/10.1109/ACCESS.2024.3383431

Nguyen, M.-D., La, V.-H., Mallouli, W., Cavalli, A.R., Oca, E.M.D., 2024. Toward Anomaly Detection Using Explainable AI, in: Sadovykh, A., Truscan, D., Mallouli, W., Cavalli, A.R., Seceleanu, C., Bagnato, A. (Eds.), CyberSecurity in a DevOps Environment. Springer Nature Switzerland, Cham, pp. 293–324. https://doi.org/10.1007/978-3-031-42212-6_10

Noorunnahar, M., Chowdhury, A.H., Mila, F.A., 2023. A tree based eXtreme Gradient Boosting (XGBoost) machine learning model to forecast the annual rice production in Bangladesh. PLOS ONE 18, e0283452. https://doi.org/10.1371/journal.pone.0283452

Ogundipe, A.O., 2024. Adaptive Load Balancing and Auto Scaling Algorithms for Resource Optimization in Distributed Microservices based Cloud Applications. Int. J. Sci. Archit. Technol. Environ. 101–111. https://doi.org/10.63680/ijsate0524111.06

Queiroz, R., Cruz, T., Mendes, J., Sousa, P., Simões, P., 2024. Container-based Virtualization for Real-time Industrial Systems—A Systematic Review. ACM Comput. Surv. 56, 1–38. https://doi.org/10.1145/3617591

Roshan, K., Zafar, A., 2022. Using Kernel SHAP XAI Method to Optimize the Network Anomaly Detection Model, in: 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom). Presented at the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), IEEE, New Delhi, India, pp. 74–80. https://doi.org/10.23919/INDIACom54597.2022.9763241

Sayegh, H.R., Dong, W., Al-madani, A.M., 2024. Enhanced Intrusion Detection with LSTM-Based Model, Feature Selection, and SMOTE for Imbalanced Data. Appl. Sci. 14, 479. https://doi.org/10.3390/app14020479

Singh, K., Kashyap, A., Cherukuri, A.K., 2025. Interpretable Anomaly Detection in Encrypted Traffic Using SHAP with Machine Learning Models. https://doi.org/10.48550/ARXIV.2505.16261

Suboh, S., Aziz, I.A., Shaharudin, S.M., Ismail, S.A., Mahdin, H., 2023. A Systematic Review of Anomaly Detection within High Dimensional and Multivariate Data. JOIV Int. J. Inform. Vis. 7, 122. https://doi.org/10.30630/joiv.7.1.1297

Vankayalapati, R.K., 2025. Composable Infrastructure: Towards Dynamic Resource Allocation in Multi-Cloud Environments. SSRN Electron. J. https://doi.org/10.2139/ssrn.5121215

Wewaldeni Pathirannehelage, Y., 2022. Analysis of centralized to federated learning-based anomaly detection in networks with explainable AI (XAI).