National College of Ireland

Ensemble-Driven Cloud Anomaly Detection Using Advanced Boosting Models and Explainable AI

MSc Research Project
MSc in Cloud Computing

Suhail Kuzhikandathil Sageer
Student ID: 22248048

School of Computing
National College of Ireland

Supervisor: Sai Emani

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Suhail Kuzhikandathil Sageer

Student ID: 22248048

Programme: MSc in Cloud Computing **Year:** 2025-2026

Module: Research Project

Supervisor: Mr. Sai Emani

Submission Due Date: 11/12/2025

Project Title: Ensemble-Driven Cloud Anomaly Detection Using Advanced Boosting Models and Explainable AI

Word Count: 8283 **Page Count** 21

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: Suhail Kuzhikandathil Sageer

Date: 10/12/2025

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):	

Ensemble-Driven Cloud Anomaly Detection Using Advanced Boosting Models and Explainable AI

Suhail Kuzhikandathil Sageer

22248048

Abstract

Cloud infrastructures produce large-scale, highly variable workloads where subtle anomalies can cause performance degradation, service downtime and financial loss. There are some traditional rule-based monitoring systems which usually fail to capture complex, multi-dimensional patterns by creating an urgent need for intelligent, data-driven detection. This study develops an ensemble-driven cloud anomaly detection framework using advanced boosting algorithms which are chosen for their ability to model nonlinear relationships, correct errors sequentially and achieve high accuracy on noisy and imbalanced datasets. Boosted classifiers were specifically focused because techniques like XGBoost, LightGBM, CatBoost, and Gradient Boosting excel at handling heterogeneous cloud features by reducing bias and variance and providing highly stable predictions. This study also includes real-time inferencing capability where the trained model has been deployed on AWS EC2 and connected with a synthetic workload stream. The system continuously monitors workload behaviour and forecasts anomaly risks which triggers notifications with the help of AWS SNS. This study also shows good results that the CrossBoost ensemble achieves 95% accuracy which is quite superior to all individual classifiers showing the strength of heterogeneous boosting in identifying rare anomalies. LIME-based Explainable AI further clarifies prediction drivers by enhancing transparency. The findings advance state-of-the-art anomaly detection by proving that multi-boosting ensembles outperform traditional ML and standalone models.

Keywords: Cloud Anomaly Detection, Ensemble Learning, Boosting Algorithms, Meta-Modelling, Explainable AI

1 Introduction

This chapter introduces the foundation of the study by outlining the background of cloud anomaly detection, the importance and motivation for adopting AI-driven approaches, the research question and objective guiding the work, and the overall structure of the report.

1.1 Background

Cloud computing has become the backbone of the current digital infrastructure and it allows computing resources and services to be delivered at scale, flexibly and at a low cost (George et al., 2023). Cloud platforms are used by organizations in various industries to provide management of applications, store huge amount of data, and facilitate real time operations. Nevertheless, the dynamic and distributed character of the cloud environments also poses considerable issues in ensuring the performance, reliability, and security (Amiri et al., 2023). Among the most significant ones is the presence of anomalies, or unanticipated deviations of

resources usage or system operation, which may result in the service degradation, downtime, or financial loss. Conventional monitoring devices tend to be reactive and will not be intelligent enough to recognize complex, multi-dimensional and time-dependent trends in real time (Keum et al., 2024). This drawback shows the necessity of sophisticated, active abnormality identification systems that are driven by artificial intelligence (AI) and machine learning (ML). Long Short-Term Memory (LSTM) networks and Bidirectional LSTM (BiLSTM) networks have demonstrated a high potential in time-series data, particularly in identifying temporal dependencies in time-series data. Using such models and supplementing them with features of contexts, one can observe anomalies more correctly and effectively. The study is aimed at creating and testing AI-based solutions to cloud anomaly detection, which will lead to more resilient, autonomous and self-optimizing cloud management systems.

1.2 Importance of the Study

The importance of this study is that it will help improve the reliability, performance and security of contemporary cloud computing systems by means of intelligent anomaly detection. With more organizations facing the need to use cloud infrastructure to deploy key applications and handle large volumes of data, unexpected resource spikes, node outages or service failures can cause huge losses in their operations and finances. There is some traditional monitoring solutions are prone to be reactive and cannot recognize complex, multivariate and time-dependent patterns in real time. In this study, the proposed approach would help to resolve these limitations by employing advanced AI and Machine Learning models to process time-series cloud trace data and identify initial indications of anomalies.

1.3 Research Question

Cloud-based infrastructures mainly experience unexpected performance degradation or service disruptions due to undetected anomalies in resource usage patterns. Traditional monitoring systems trusts on reactive mechanisms that fail to capture complex temporal dependencies which leads to delayed responses and reduced reliability.

"What combination of boosting techniques can most effectively detect cloud resource anomalies in large-scale cluster environments while ensuring high accuracy, low latency, and strong interpretability through Explainable AI?"

1.4 Research Objective

To develop a strong cloud anomaly detection framework using advanced ensemble-based machine learning techniques using the Google Cluster dataset to accurately identify irregular resource-usage patterns while ensuring high performance, scalability, and interpretability through model evaluation metrics and Explainable AI methods.

1.5 Motivation

Cloud environments manage large-scale infrastructure with thousands of machines and services. Detecting anomalies in real-time is very important to avoid system crashes, improve up time and secure good resource utilization. Manual monitoring is insufficient due to scale and complexity by making AI-driven solutions a practical necessity.

1.6 Structure of the Study

This report is organised into seven chapters. Chapter 1 introduces the background, motivation, research question, and objectives of the study. Chapter 2 reviews existing literature on machine learning and deep learning-based anomaly detection in cloud environments. Chapter 3 details the methodology, including dataset description, preprocessing, and feature engineering. Chapter 4 explains the system architecture, ensemble design, and research specification. Chapter 5 discusses implementation across Google Colab and AWS. Chapter 6 evaluates model performance using confusion matrices, latency, throughput, and baseline comparison. Finally, Chapter 7 concludes the study and outlines future research directions.

2 Related Work

This chapter reviews the foundational concepts and prior research relevant to cloud anomaly detection. The chapter further discusses ML and DL methods capable of modeling high-dimensional temporal patterns showing their strengths and limitations.

2.1 What is Cloud Anomaly Detection?

Cloud anomaly detection refers to the act of detecting atypical or deviant trends in cloud-based systems, apps, or information that is not typical (Hagemann and Katsarou, 2020). In a cloud computing scenario where large volumes of data are produced in just a few seconds, anomaly detection (Islam et al., 2025) will be of paramount importance to sustain operations, secure the system, and avoid possible system breakdown or even a cyberattack. The anomalies in the cloud may take different types like the abrupt increase in the network traffic, unauthorized access to data, unexpected downtime of the systems, or unusual resource consumption (Oluwafemi Esan, 2024). These deviations are usually indicative of such problems as intrusions, misconfigurations, or service failures, or malicious actions. Cloud anomaly detection is the automatic analysis of the data streams using statistical analysis, machine learning, and artificial intelligence to detect such anomalies in real-time (Zhang et al., 2025). Conventional systems based on rules commonly do not adequately react to the dynamic and distributed character of the cloud environments; therefore, the contemporary solutions apply sophisticated algorithms that are based on the past data to determine the behavioural guidelines. Upon the detection of the deviations, automatic notifications can be produced to initiate an immediate inquiry or instead mitigation measures. This active identification assists companies in determining the dependability of their services, optimize the use of cloud resources, and have a better cybersecurity protection. Besides, anomaly detection is used by compliance and operational governance to ensure transparency and accountability in the multi-tenant cloud infrastructures. As hybrid and multi-cloud systems are more embraced, anomaly detection has become an essential part of cloud computing to aid in decision-making by providing smart insights via ongoing monitoring. In a sense, cloud anomaly detection (Xin et al., 2023) serves as an online security system, an intelligent watch tool that not just points at possible threats or unproductive actions ahead of time but also boosts the overall resilience, trust and efficiency of the cloud system, in the current data-driven, highly connected ecosystem of clouds.

2.2 Machine Learning for Anomaly Detection in Cloud Computing

Machine Learning (ML) has become an effective means of improving anomaly detection in the cloud computing infrastructure, where traditional security mechanisms are not adequate due to

highly complex, large-scale, and dynamic operations. Different research have examined the different ML-based strategies in order to enhance detection rates, reduce the occurrence of false alarm and maximize the computational efficiency in order to achieve optimal results. A time series-based intrusion detection model was proposed in (Al-Ghuwairi et al., 2023) based on a collaborative feature selection (FS) model, which operated on the Facebook Prophet prediction framework. This strategy was able to reduce the number of predictors (70) to 10 and enhance performance measures (MAE, RMSE and DTW), which indicates that Prophet is effective in predicting early intrusions, but is constrained by the fact that the model assumes trends. In (Sherubha et al., 2023), a smart Network Intrusion Detection System (NIDS) was created based on an auto-encoder to select the features and a Naive Bayes classifier to classify the features. The model reached 93% accuracy and True Positive Rate of 0.99 which was better than the current algorithms but limited by the fact that the Naive Bayes model assumes the independence of the features. Likewise, (Alghushairy et al., 2024) presented an anomaly-based Network Outlier Detection System (NODS) that is a combination of Support Vector Machine (SVM) and Gaussian Naïve Bayes (GNB) using Genetic Algorithm to optimize the parameter values. On NSL-KDD and CICIDS2017, the SVM-based model performed better with an AUC of 0.98 and better detection speed but also added more computational complexity to the Genetic Algorithm.

To further improve the security of the cloud, (Arunkumar and Kumar, 2023) was able to propose a hybrid Support Vector Machine-Extreme Learning Machine (SVM-ELM) supported by Gannet Optimization Algorithm (GOA) to detect intrusion. The model achieved a better classification and the time to do training was under 400 ms in several data sets, but it was not scalable in real-time systems. In solving the problem of insider threats, (S et al., 2023) developed a two-layer model that integrated NearMiss2 (NM-2) under sampling with One-Class SVM to resolve the issue of the class imbalance (CIP). This method delivered a recall of 100 percent and an accuracy of 82.46 percent, which is very powerful in insider threat detection, but the risk of loss of information through under sampling was a weakness. In the case of IoT-integrated cloud environments, (Sahaaya Arul Mary et al., 2025) applied the K-Nearest Neighbors (KNN) classifier using a 5G and cloud computing environment in real-time to detect anomalies. The system had an 98.4% accuracy, but it was challenged by the scalability of KNN to large datasets because of the computational intensity of the method. Finally, (Jiang et al., 2023) suggested an adaptive ensemble random fuzzy (AERF) algorithm, which requires the fusion of fuzzy rule-based random sampling and weighting in real-time in order to deal with the imbalance and continuous random clouds. The AERF had performance as high as 20 percent higher AUC and F1 scores along with 50 percent reduced training time in comparison with the existing models and random sampling might do away with the uncommon anomalies. In aggregate, these studies show that ML-based anomaly detection in cloud computing contributes to a great extent in improving detection accuracy, scalability, and system resource use, but the problem of scalability, real-time introducing responsiveness, and data disequilibrium remains.

2.3 Deep Learning for Anomaly Detection in Cloud

Deep learning has become the new revolution in identifying anomalies in cloud computing and other intelligent systems in which it can give flexible and precise solutions in determining anomalies in multifaceted, high-dimensional data. The authors in (Ahmed et al., 2025) investigated the use of advanced deep learning models including RNN, CNN, and LSTM to enhance security of a cloud data centre and reported a 0.90 ROC AUC score and suggested the possibility of LSTMs to detect sustained threats but with a high computation cost. Applying

this point of view to the case of the IoT environment, (V. and A., 2023) suggested a hybrid MarkovLSTM model of real-time anomaly detection in smart devices, which successfully detects the false data and outliers of sensor networks with 96.03% detection accuracy and 92.48% training accuracy, which is better than conventional algorithms such as KNN. In part, (Girish and Rao, 2023) concentrated on cloud-based OpenStack systems, proposing Stacked and Bidirectional LSTM models to predict system anomalies with the help of information gathered by the means of Collected, where the detection rates were 94.61 (training) and 93.98 (testing), though, they had issues with the model scalability and interpretability. It was later improved upon in (Wang, 2024) to introduce a lightweight CNN-BiLSTM-attention-based model that is combined with a Wasserstein GAN (WGAN) to manage data imbalance and improve anomaly detection using cloud-edge architecture with a benchmark accuracy of 0.974, 0.925, and 0.953 on NSL-KDD, UNSW-NB15 and CIC-IDS2018, respectively. Finally, (Natha et al., 2025) took the field a step further, to video-based anomaly detection that proposed Composite Recurrent Bi-Attention (CRBA) model, which integrated DenseNet201, BiLSTM and attention to identify abnormal activities in surveillance video with 92.2% accuracy. Taken together, these works point to the increasing importance of deep learning in the field of anomaly detection in various areas such as cloud, edge, and IoT and represent impressive accuracy and scalability. Nevertheless, there are still difficulties concerning computational overhead, generalization of models and real-time scaling with the most lightweight, interpretable, and energy-efficient deep learning architecture being in the spotlight of further anomaly detection in cloud ecosystems.

Table 1: Summary of Reviewed Studies on ML & DL for Anomaly Detection in Cloud Computing

Study	Proposed Approach	Methods & Techniques Used	Key Results / Findings	Limitations
(Al-Ghuwairi et al., 2023)	Collaborative Feature Selection with Facebook Prophet model	Time series analysis, anomaly detection, causality tests, Prophet forecasting	Reduced predictors from 70→10, improved MAE, RMSE, DTW, and reduced training time by 85%	Limited by Prophet's assumptions on trend and seasonality
(Sherubha et al., 2023)	Auto-encoder + Naïve Bayes-based NIDS	Auto-encoder for feature selection, Naïve Bayes classifier	Achieved 93% accuracy, TPR=0.99, FAR=0.3, outperforming other ML models	Naïve Bayes assumes feature independence, limiting correlated data handling
(Alghushairy et al., 2024)	Anomaly-based Network Outlier Detection System (NODS) using SVM and GNB	PCA, CFS for feature selection; SVM & GNB with Genetic Algorithm optimization	SVM AUC=0.98, GNB AUC=0.88; low false alarms and detection time	High computational cost due to Genetic Algorithm optimization

(Arunkumar and Kumar, 2023)	Gannet Optimization Algorithm-based hybrid SVM-ELM	GOA for optimal feature selection; hybrid SVM-ELM model	High precision, recall, F-measure; training time ~389–399 ms	Computational complexity and limited scalability in real-time
(S et al., 2023)	Double-layer architecture with NM-2 and One-Class SVM	Data sampling (NearMiss2), anomaly detection (One-Class SVM)	Recall=100%, F-score=78.72%, Accuracy=82.46%	Undersampling may remove valuable majority class data
(Sahaaya Arul Mary et al., 2025)	KNN classifier with 5G and cloud-based IoT data analysis	KNN for classification; cloud-based ML; 5G for real-time data	Achieved 98.4% accuracy, outperforming other models	High computation cost due to distance calculations in KNN
(Jiang et al., 2023)	Adaptive Ensemble Random Fuzzy (AERF) algorithm	Random fuzzy rule-based method with dynamic weighting	10–20% higher AUC/F1; 50% faster training time	Random sampling may overlook rare anomalies
(Ahmed et al., 2025)	Deep Learning Models (RNN, CNN, LSTM)	Compared multiple DL models on cloud data; LSTM performed best	LSTM achieved ROC AUC 0.90	High computational cost and limited scalability
(V. and A., 2023)	Hybrid Markov–LSTM Model	Used real-time DHT sensor data (temperature, humidity); anomaly detection via Markov + LSTM	Detection accuracy 96.03%, training accuracy 92.48%	Scalability issues and energy constraints on edge devices
(Girish and Rao, 2023)	Stacked & Bidirectional LSTM	Used Collectd to gather 10-feature dataset; binary classification via LSTM	Training accuracy 94.61%, testing accuracy 93.98%	Computational overhead and limited interpretability
(Wang, 2024)	Lightweight CNN–BiLSTM–Attention + WGAN	WGAN for data synthesis; CNN for spatial, BiLSTM for temporal, Attention for feature weighting	Accuracies: 0.974 (NSL-KDD), 0.925 (UNSW-NB15), 0.953 (CIC-IDS2018)	Synchronization and privacy issues across edge nodes
(Natha et al., 2025)	Composite Recurrent Bi-Attention (CRBA) Model	Combined DenseNet201 + BiLSTM + Multi-Attention for spatiotemporal analysis	Detection accuracy 92.2% on UCF & RAD datasets	Performance affected by lighting, occlusion, and camera angles

3 Methodology

This chapter outlines the methodology used to build the anomaly detection system, covering the research problem, dataset details, and all preprocessing steps. It also explains feature-importance selection which ensures the data is optimized for accurate boosting-based anomaly detection.

3.1 Research Problem

The current cloud computing systems are at a huge scale, with millions of workload events in the processes of CPU, memory, scheduling, and resource allocation. Providing cloud services that can handle these dynamic workloads without performance stability is still a serious challenge that cloud providers face. Historical system monitors are still dynamic and mostly based on fixed thresholds and rule-based notifications, which tend to miss the complex workload variations or subtle anomalies caused by resource contention, resource schedule variations, unexpected usage bursts, or even inefficient job execution. The constraints of traditional monitoring introduce reliability lapses, resulting in the reduction of the performance, service disruptions, and optimal use of resource usage as cloud workloads change to be more heterogeneous, bursty, and uncertain. The necessity of a smart, data-driven tool that would be able to learn behavioural pattern of the cloud and detect anomalies in real-time has thus become an urgent matter. To solve these issues, this study takes a machine learning-based model, which will analyse the Google 2019 trace Cluster with boosting models and meta- ensemble architecture by utilizing boosting models to detect anomalies correctly, scalable, and interpretably, and to stabilize operations within cloud computing.

3.2 Dataset Description

The dataset applied in this study is the Google 2019 Cluster Trace, which has been taken in Kaggle, and it is a collection of workload activity details on eight production Google Barg compute clusters in the month of May 2019. This dataset gives a detailed account of the large-scale cloud operations, as it records all job submissions, scheduling actions, resource requests, and resource usage activities that take place in the clusters. It builds upon the popular 2011 Google trace by adding new machine architectures, new workload trends, and more detailed metadata, an evolution of the cloud computing systems over the years. One major improvement over this version is that sample histograms of CPU usage are now shown every five minutes, allowing a more realistic model of workload variation than a single point sample will allow. The data further contains alloc sets (shared resource allocations to multi-job workloads), job-parent links between distributed jobs like MapReduce and detailed scheduling. These characteristics render the dataset very useful in identifying anomalies, performance modelling and variability in workload. Notably, the traces do not include user-specific information, so it is practical, yet the privacy is guaranteed. The presence of the dataset on such platforms as Kaggle and Google BigQuery enables an efficient access, querying scaled, and advanced machine learning analysis without the need to use large local storage.

3.3 Data Cleaning

The data cleaning phase was involved in the preparation of the Google Cluster Trace data to be able to detect anomalies reliably by addressing the missing entries, unnecessary data, and deriving the desirable information out of the complex attributes. Some of the fields included

nested JSON like strings that showedCPU and memory usage that had to be custom parsing. It was replaced by a special purpose split_cell() function that was used to decipher numeric CPU and memory values by breaking up text patterns, isolating digits, and converting them into floating-point features representing NaN where bad data was present or non-numeric data. This operation was performed on various columns like average usage, max usage and resource requests to obtain the structured attributes like average_cpu, maximum memory and the request of the resource in the form of CPU max memory and resource request. Many of the high-null and non-essential columns such as scheduling metadata, random usage samples, distribution histograms, and constraint fields were removed in order to reduce noise and enhance efficiency of the model.

3.4 Data Preprocessing

The preprocessing phase of data prepares the cloud trace data to be used by machine learning with trusted reliability to address inconsistencies in numbers, categorical attributes, and class skewness, and normalizes the features to the range. The combination of these steps guarantees that the dataset turns out to be formatted, constant and best suited to detect anomalies well in all the boosting and ensemble models.

3.4.1 Outlier Handling

The use of outlier handling was a necessary preprocessing stage because cloud workload traces are highly volatile and bursty as shown in Figure 1. Various numeric characteristics such as assigned memory, page-cache memory, CPU utilization, and requested resources had large values caused by temporary workload bursts or some variation in measurement intervals. The analysis of detecting these anomalies was conducted by first carrying out a boxplot inspection to show the distribution and indicate the deviations above the standard deviations. Each of the selected columns was then subjected to the Interquartile Range (IQR) technique. Q1, Q3, and IQR values were calculated on each feature and then any record that was below the lower limit or above the upper limit (Q1 -1.5 x IQR or Q3 + 1.5 x IQR) was eliminated. This process of filtering was repeated until there were no further outliers left which provided the clean dataset stability. The index was also processed again to ensure structural consistency. This removal of these extreme values resulted in a more trusted dataset to run machine learning and a stronger model.

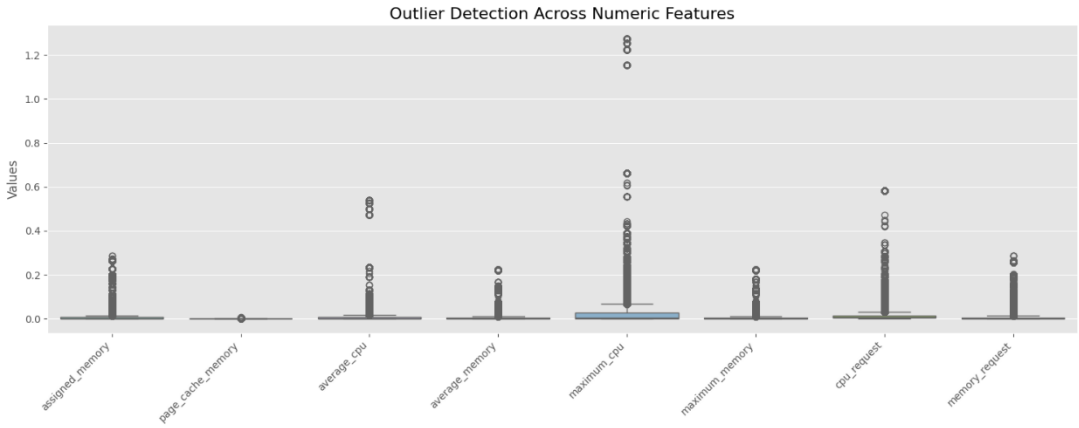

Figure 1: Outlier detection

3.4.2 Label Encoding

Categorical attributes like scheduling class, type of collection, user identifiers, and names of collection are common attributes that can be found in cloud trace datasets and are not directly interpretable by machine learning algorithms. Label Encoding was used on all the appropriate categorical columns to transform these non-numeric fields into meaningful numerical representations. All the distinct categories were systematically coded to an integer, maintaining the difference between classes and being compatible with boosting-based models. This method is computationally fast, and it is especially applicable to high-cardinality attributes of cloud workload logs. The categorical data were changed into numerical ones to make the dataset fully machine-readable, which enabled the model to learn scheduling behaviour patterns, workload types and priority classes. Contrary to one-hot encoding, Label Encoding did not expand the dimensionality, and the dataset became small and efficient to train and infer. Figure 2 shows correlation matrix which does have some numeric features only.

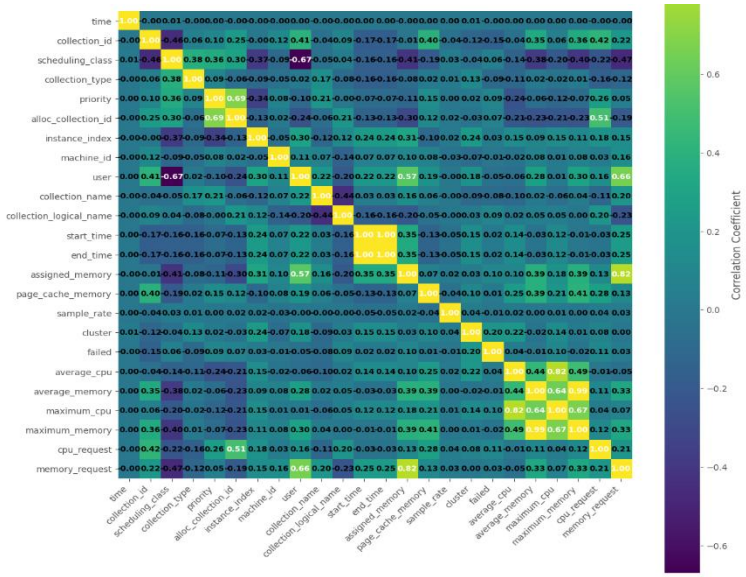

Figure 2: Correlation Matrix

3.4.3 Handling Imbalanced Data (SMOTE)

The data used in the analysis had a huge skew between normal and anomalous events, and therefore, this could skew the model to the majority classes, and the performance of anomaly detection would be low. To counter this, the SMOTE (Synthetic Minority Over-sampling Technique) was used to reassign the data set. SMOTE uses minorities through interpolation of the existing minority data to produce synthetic minorities without noise to the underlying structure and distribution of the data. Such a strategy will make the training set more homogeneous in terms of the percentage of classes, without merely replicating records of minorities. The labels (y) and features (X) were inputted to the SMOTE algorithm which gave resampled sets with a higher proportion of anomalous events. This increased learning capacity of the models to learn subtle patterns related to anomalies, precision and recall and minimised false negatives which is essential in cloud computing settings where anomalies detected and not are likely to cripple performance, create bottlenecks in resources or lead to failures. All in all, SMOTE enhanced reliability of detection in various workload conditions. Figures 3 and 4

shows the SMOTE based balancing process, showing how synthetic minority samples reshape the class distribution to reduce severe imbalance in anomaly labels.

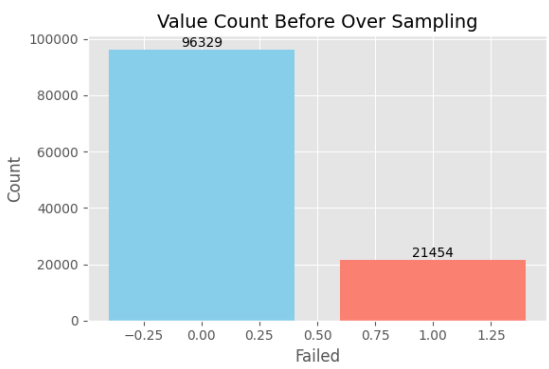

Figure 3: Before over sampling

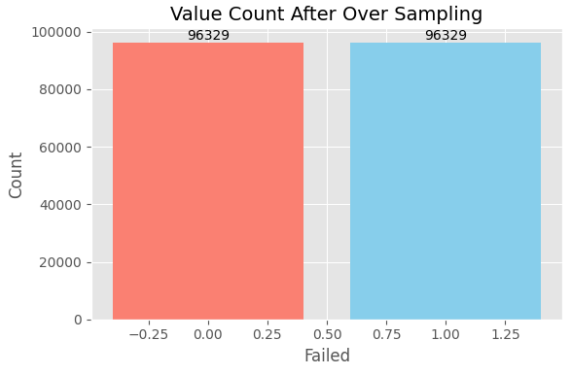

Figure 4: After over sampling

3.4.4 Data Scaling

The data scaling was used to make sure that all the numerical features played an equal role in the machine learning model and to prevent the dominance of the attributes with bigger values range. Because the dataset used includes heterogeneous values, such as CPU usage fractions and memory allocations in absolute units, it can be stated that MinMaxScaler was applied to all of them to normalize them in a 0 to 1 range. This change maintained the original distribution form but decreased the variability across the attributes, which is important especially to those ensemble algorithms that are sensitive to scale variations. The transformed data after scaling was reassembled into a standardized DataFrame to preserve the integrity of columns to be used later. Scaling was also useful in the convergence of models during the training process and stability in distance-based calculation in boosting learners. Moreover, it standardized the value of resource usage into a standardized representational form which allowed the anomaly detection models to be efficient in the capture of subtle workload behaviour deviations. This preprocessing phase gave a normalized, well-balanced dataset which was highly optimized to train high performance, classification models.

3.5 Feature Importance

Feature importance analysis was conducted to identify the most influential attributes contributing to cloud anomaly detection as shown in Figure 5. A Decision Tree classifier was trained on the oversampled dataset (sm_x, sm_y), and its feature importance scores were extracted to quantify each variable's predictive contribution. These importance values were combined with feature names and sorted to create a ranked list. The top 15 features were then selected to form a reduced feature subset, ensuring that only the most relevant predictors were used during model training. This dimensionality reduction improved model efficiency, reduced noise, and enhanced both training speed and overall anomaly detection performance.

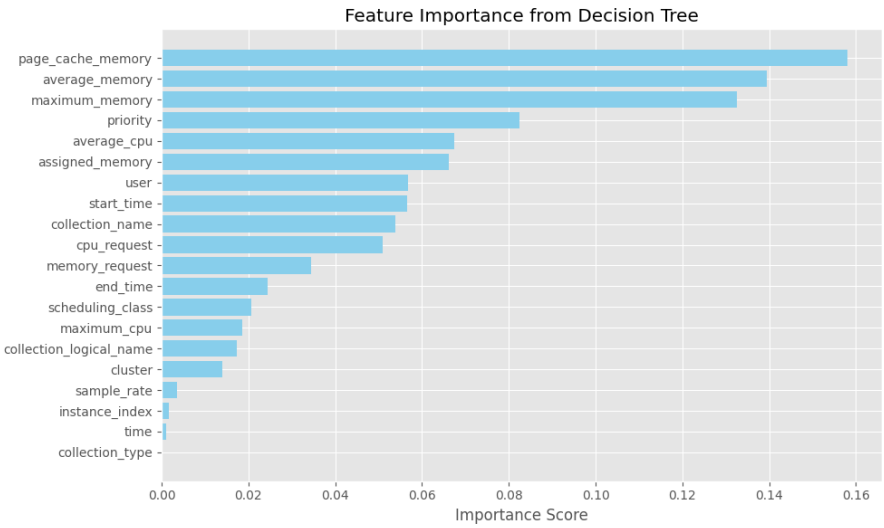

Figure 5: Feature Importance

4 Design Specification

This chapter presents the complete design specification of the proposed cloud anomaly detection system. It outlines the system architecture, the CrossBoost meta-ensemble design, and the research specification guiding performance-optimized anomaly detection

4.1 System Architecture and Components

The figure 6 demonstrates cloud anomaly detection system end-to-end workflow. The first stage involves acquiring the Google Cluster data set, which is then loaded, cleaned, and fully preprocessed, including outlier removal, coding and scaling. Importance of features is calculated in a random forest and then the data are split into training. The main structure is the CrossBoost Meta-Ensemble that is an integration of several boosting base learners and a LightGBM that is used as the meta-learner so that it can achieve better accuracy and strength. The trained model is also evaluated in terms of confusion matrices and classification measures, as well as performance analysis. Lastly, the pipeline is finished with explainability with the deployment of LIME and AWS with EC2, Cloud9, and S3.

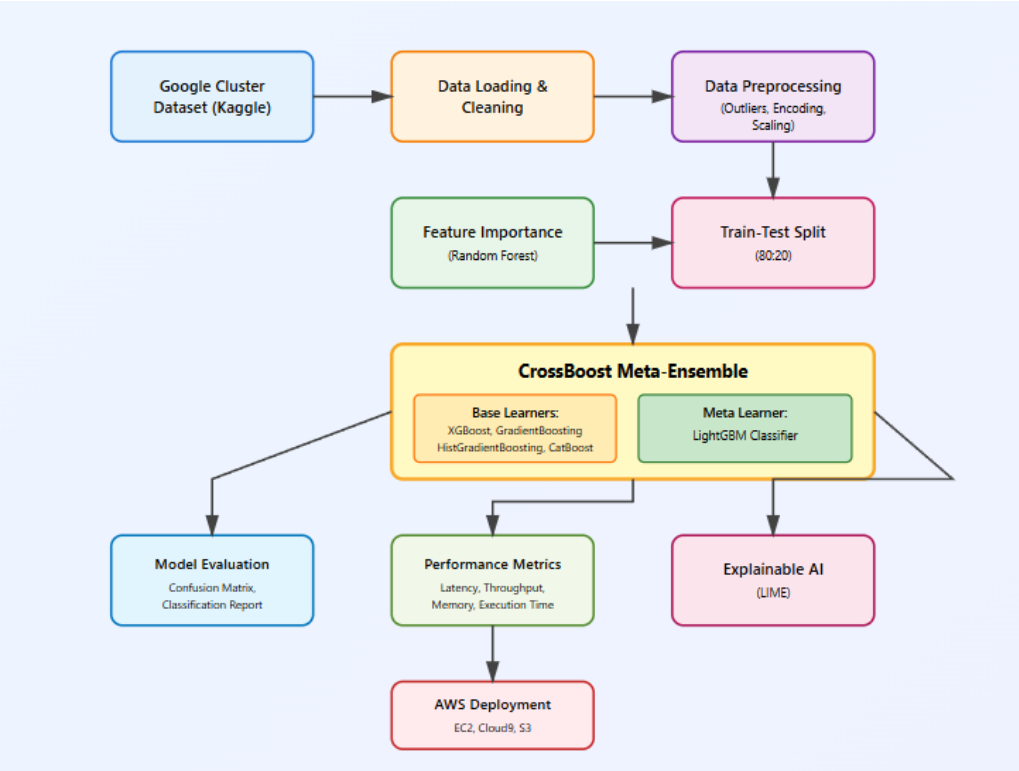

Figure 6: System Architecture

4.2 Research Specification: Anomaly Detection and Cloud Performance Optimisation Approach

The research specification focuses on building an optimized anomaly detection pipeline, which can handle large-scale cloud workload traces without allowing errors and at low computational costs. The proposed design incorporates numerous boosting models and a LightGBM meta-learner to detect the intricate behavioural variation in CPU, memory, and job-level trends in contrast to threshold-based monitoring systems that do not react well to variations in workloads. The pipeline of the system works with the Google Cluster Trace data and includes a preprocessing, feature engineering, ensemble learning and model explainability in a structured format. This is because of feature prioritization, which makes sure that only the most valuable attributes are utilized, and the model is more efficient and less noisy in its predictions. Combining complementary learners within the CrossBoost ensemble makes it more robust and the system is able to identify rare anomalies more effectively. The design is further enhanced with performance optimization by reducing redundancy of data by minimizing it, using vectorized computing and placing constraints on the model depth. This combined specification has allowed it to be scaled, have detection stability and be ready to be used in changing clouds.

4.3 Justification for Choosing Boosting Algorithms

The choice of boosting algorithm was due to its ability to detect complex and subtle anomalies in cloud workloads through sequential correction of past errors. Unlike single classifiers, boosting has the benefit of integrating many weak learners into a powerful, highly precise model that can handle noisy, imbalanced and high-dimensional data- another face of complexity in cloud trace data. XGBoost, LightGBM and CatBoost are some of the methods that are fast in computation, have efficient memory usage, and good generalization ability,

which is suitable in the large-scale cloud setup. The fact that they can minimise bias and variance make them stable and reliable in detecting anomalies, even in cases where workload patterns are dynamically changing.

4.4 Cloud Infrastructure

The AWS cloud computing infrastructure of the anomaly detection system is a set of provisions aimed at providing scaled computing, secure storage, and a smooth run of the ML workflow. Amazon S3 is the main storage level of raw traces, processed datasets, and trained models secured by bucket policies and encryption. Model training and batch inference on EC2 instances are controlled by security groups. Cloud9 is the development environment which allows to configure Python libraries and data pipelines. The infrastructure makes all calculations take place within AWS, and it minimizes data transfer risks as well as the ability to provide big workload analytics with significant efficiency and reliability.

4.5 Detection Flow

The process of anomaly detection starts when the processed workload data is uploaded in S3 into the EC2-based ML engine. The trained boosting models utilize the CPU usage, memory usage, schedule of task execution and histogram of their usage to classify each record as a normal or abnormal point. The prediction logs are written and stored in S3 together with assessment reports. EC2 makes inference and creates detection outputs automatically when new data concerning cloud workloads is received. This architecture can be used to generate consistent, real-time detection of anomalies and has a traceable and completely logged prediction cycle, which can be used in a large-scale cloud deployment.

5 Implementation

This chapter outlines the complete implementation of the proposed anomaly detection system. It describes the dual development environment, explainable ai, cloud deployment and implementation of ml models.

5.1 Development Environment

A hybrid development combining Google Colab and AWS Cloud9 was used to implement the same. Preliminary experimentation and rapid prototyping, as well as model testing with the help of a given GPU, were done with Colab, whereas Cloud9 provided an integrated AWS environment, which is linked to an EC2 instance to perform scaling processing. Preprocessing and training models were installed with such essential libraries of Python as NumPy, Pandas, Matplotlib, Seaborn, Scikit-Learn, XGBoost, LightGBM, CatBoost, and Imbalanced-Learn. Such extra utilities as psutil to monitor memory and boto3 to interact with Amazon were added. This setup guaranteed that it was scalable, reproducible and that the Google Cluster Trace dataset could be handled efficiently.

5.2 Explainable AI Implementation Using LIME

In order to achieve greater interpretability and transparency of the CrossBoost anomaly detection model, the system incorporates an Explainable AI module with the LIME (Local Interpretable Model-Agnostic Explanations) framework. The localized and human readable

explanations of the individual predictions in LIME are critical in cloud anomaly detection that requires the cause of abnormal behaviour just as much as detecting it. The LimeTabularExplainer was configured with the processed training data, names of the features, and the class labels which ascertained that the explanations captured the structure of the transformed data. The stacked ensemble was modified to have a custom prediction wrapper, which enabled LIME to produce explanations of any input instance in the form of probabilities. As an example, one of the items in the test set was chosen and inputted to the explainer, and the explainer generated a ranked list of features that had a positive or negative contribution towards the anomaly classification. Such outputs bring to focus powerful values like CPU consumption, memory changes and workload scheduling trends.

5.3 Implementation of ML Models

The implementation of the machine learning model is cantered on the development of a highly effective framework of anomaly detection based on boosting and five independent classifiers, as well as the use of meta-ensemble architecture. GradientBoostingClassifier, HistGradientBoostingClassifier, XGBClassifier, LGBMClassifier, and CatBoostClassifier are the most basic learners that were pre-initialized with optimized hyperparameters so that the latency can be low and the model complexity can be controlled. Both models were trained with an 80:20 train test split on the processed dataset and, therefore, can learn various behavioural patterns in the cloud workload traces independently. These classifiers were combined to form a CrossBoost Meta-Ensemble, which was trained with a StackingClassifier in LightGBM as the meta-learner to improve the stability of the detection. The meta-model combines prediction probabilities produced by the base models and learns more high-level patterns which enhance generalization and minimize misclassification. This multiple-layered model implementation guarantees a solid detection of anomalies in varying cloud workloads, and it is more accurate, precise, and dependable than single models.

5.4 AWS Cloud Deployment

AWS Cloud9 environment have used to deploy the cloud anomaly detection system. The Cloud9 IDE connected with an EC2 instance that offers computational resources to preprocess and train models and evaluate them. The EC2 instance has related to the Cloud9 environment that is the main computational engine in the anomaly detection system. The EC2 instance is having the ability to support scalable training of models, secure communication with S3 and consistent performance in the runtime, which is important to the cloud-based implementation of the system.

The Amazon S3 dashboard created a specific storage bucket on which it will use in the implementation pipeline. This bucket is an expandable, robust, and secure layer of cloud storage that can be used to upload datasets, store processed files and store the outcomes of the experiments that are generated in the process of anomaly detection. It plays a significant role in making sure that data is reliable when needed by the downstream processes like model training and evaluation on AWS. S3 will arrange data in form of buckets, enables region configuration, and integrates with other AWS systems such as EC2 and Cloud9, creating a smooth end-to-end workflow in the cloud environment.

5.5 Real-Time AWS Deployment and Alerting Mechanism

Integration of the trained anomaly detection model into a real-time execution pipeline running on an AWS EC2 was a part of the implementation. Synthetic workload data were sent with constant time delays to simulate real time activity in the cluster. The deployed model used the

data to make inferences and come up with predictions on the abnormal workload behaviour and failure of the systems. The pipeline was also linked to AWS SNS so that it could automatically send an alert through email when the predicted anomaly was above the set failure threshold. This configuration illustrates the way that the model can be used within a cloud-native monitoring workflow, akin to the AWS CloudWatch behaviour, to constantly evaluate the CPU utilization, workload load trends, and predict failure states as shown in Figure 7 and Figure 8.

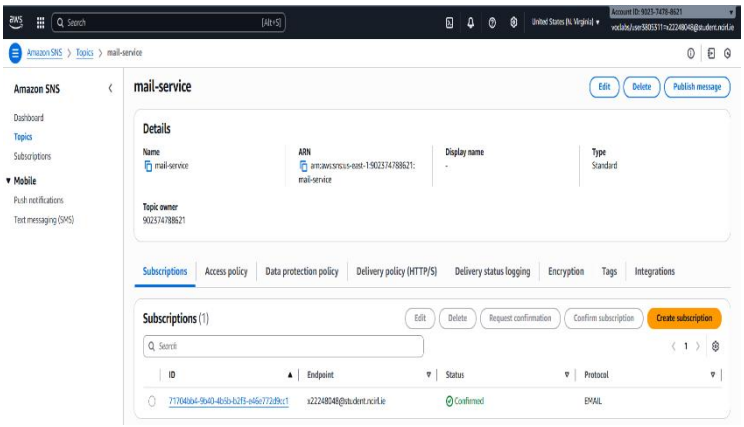

Figure 7: AWS SNS Topic and Email Subscription for Automated Alerting

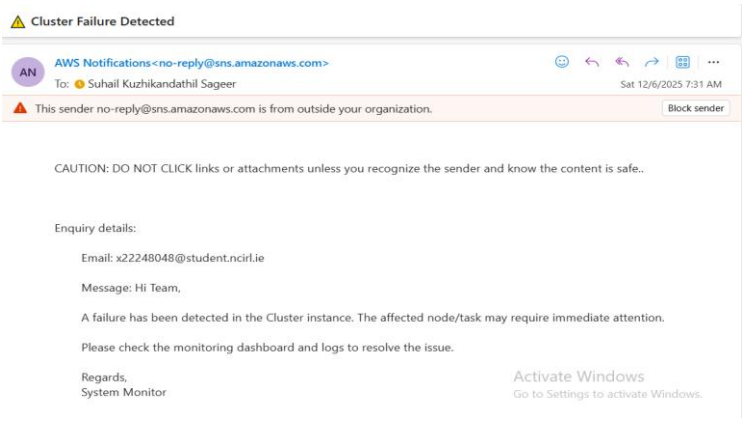

Figure 8: Cluster Failure Alert Email Generated from Real-Time EC2 Inference via SNS Notification

6 Evaluation

This chapter presents the evaluation of the proposed anomaly detection system through three experiments. The chapter also includes a baseline comparison showing how my proposed study is better than baseline study.

6.1 Experiment 1: Confusion Matrices

As shown by the confusion matrix, the Gradient Boosting Classifier can distinguish the majority of normal and anomalous events with a very few misclassifications as depicted in Figure 9. This indicates that the model has a high capacity to adapt to nonlinear workload traces in cloud traces. Its equalized error distribution implies steady generalization which the accuracy and its applicability of this method to baseline anomaly detection. The HistGradientBoosting

confusion matrix indicates that there are more misclassified anomalies than other models as indicated in Figure 10. Although the model continues to be effective in capturing overall patterns, the use of histogram in the splits does not seem to be effective in capturing delicate patterns of anomalies. This is why its macro metrics are lower, indicating that the model does not go as accurate but as fast and computational with large datasets.

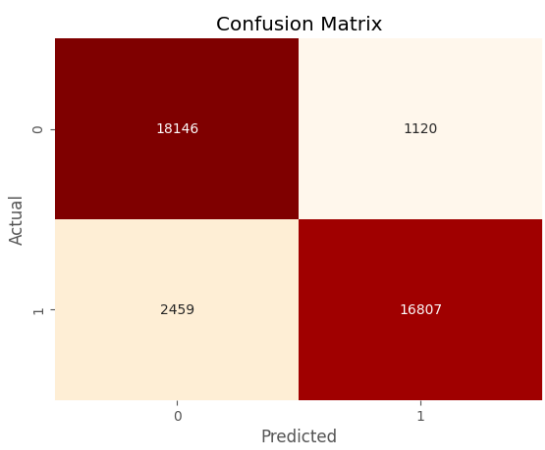

Figure 9: Gradient Boost Classifier

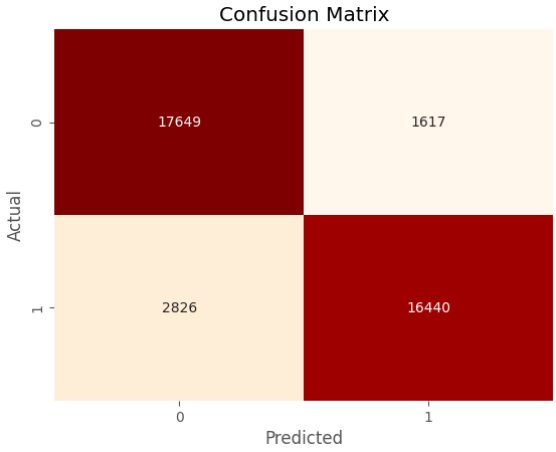

Figure 10: Hist Gradient Boost Classifier

The XGBoost confusion matrix shows that it has a good detection ability with most normal and anomalous points being properly classified with few false negatives as indicated in Figure 11. Its tree boosting structure is gradient-based, and it is capable of highly representing workload variability. This demonstrates why this model is with high precision and recall and this proves that XGBoost is among the most consistent standalone classifiers in the experiment. According to the LightGBM confusion matrix, the performance is moderately good, and the number of misclassifications is higher than in XGBoost or CatBoost. Despite being able to work well with large-scale numeric features, the leaf-wise splits in LightGBM can lead to a small overfitting of minority anomaly samples as Figure 12 demonstrates. The matrice supports its mid-range evaluation measures, which are as good as in terms of efficiency but lower in terms of anomaly sensitivity as compared to more powerful models.

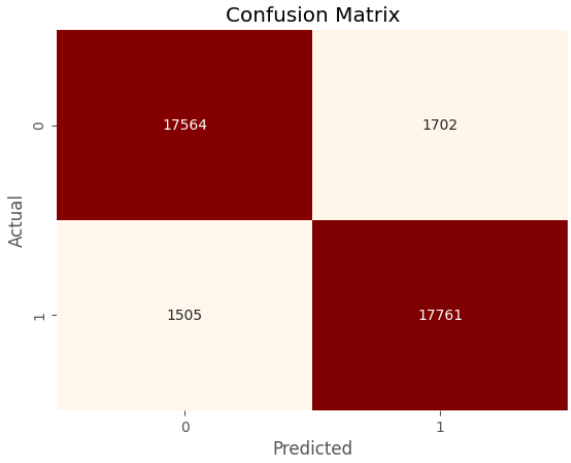

Figure 11: XGB Classifier

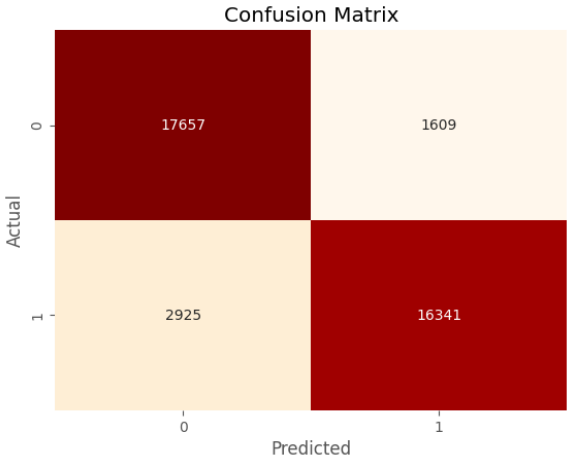

Figure 12: LGBM Classifier

The CatBoost confusion matix has a high and equal classification of normal and anomalous data with minimal false positives and false negatives as in Figure 13. Its categorical cloud-trace feature treatment characteristic features ordered boosting which makes it effective in delivering

its predictions consistently. This performance is warranted by its high-ranking accuracy/recall and similar to its performance on heterogeneous data. The confusion matrix of the ensemble presents the best classification accuracy, and very few misclassified samples are presented in Figure 14. The meta-learner is able to combine the predictions of many boosting models thereby covering a wide range of anomaly patterns and overcoming the weaknesses of a particular model. This explains its better performance metrics, and this proves that stacking has a great influence on the reliability of anomaly detection.

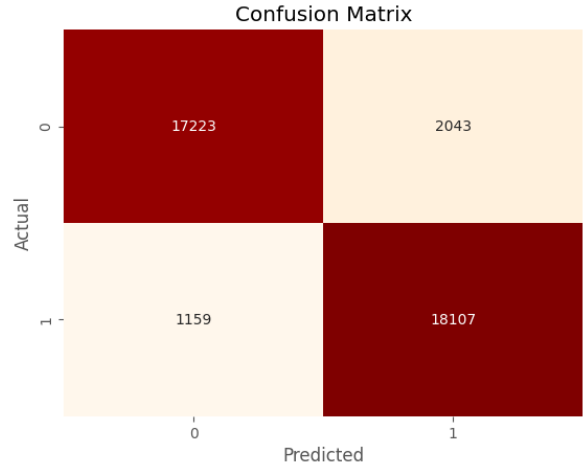

Figure 13: Cat Boost Classifier

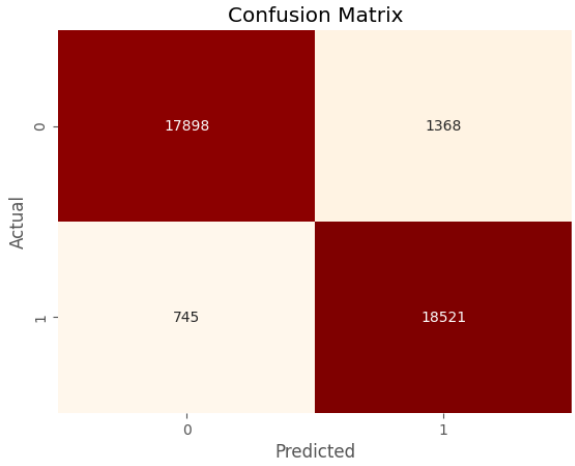

Figure 14: Stacking Classifier

6.2 Experiment 2: Latency and Throughput Analysis

This experiment evaluates the computational efficiency of the boosting models by examining latency per prediction and throughput. Latency represents the time taken to classify a single instance, while throughput measures the number of predictions processed per second—critical for real-time anomaly detection in cloud environments. The results indicate that the HistGradientBoostingClassifier achieves the lowest latency (1.28 ms), making it highly efficient for rapid inference. XGBoost shows comparatively higher latency (4.56 ms) but delivers one of the highest throughput values due to its optimized histogram-based execution. LightGBM and CatBoost provide balanced performance with moderate latency and strong throughput, aligning well with real-time cloud monitoring requirements. As expected, the CrossBoost Meta-Ensemble exhibits the highest latency (9.70 ms) because it aggregates predictions from multiple base learners, yet it offers significantly improved prediction stability and accuracy as shown in Table 2. This trade-off demonstrates that while ensembles are computationally heavier, they enhance overall robustness, making them suitable for scenarios where accuracy is prioritized over speed.

Table 2: Latency and Throughput of All Models

Model	Latency (ms)	Throughput (pred/sec)
GradientBoostingClassifier	1.4411 ms	3,348,629.84
HistGradientBoostingClassifier	1.2819 ms	1,322,966.59
XGBClassifier	4.5656 ms	3,168,485.14
LGBMClassifier	1.5278 ms	2,710,159.17
CatBoostClassifier	1.6980 ms	1,351,408.33
CrossBoost Meta-Ensemble	9.7018 ms	309,846.93

6.3 Experiment 3: Model Performance Evaluation

This experiment analyses the predictive performance of all boosting-based models to identify the most effective classifier for cloud anomaly detection. The evaluation metrics—Accuracy, Macro Precision, Macro Sensitivity, and Macro F1—provide a holistic understanding of each model's capability to handle imbalanced workload patterns within the Google Cluster Trace dataset. Among the individual models, XGBoost and CatBoost perform notably well, each achieving 0.92 across all metrics, reflecting their ability to learn complex resource usage patterns as shown in Table 3. However, the CrossBoost Meta-Ensemble outperforms all base learners, achieving 0.95 across every metric. This improvement is attributed to the stacking mechanism, where LightGBM aggregates predictions from multiple models, leveraging complementary strengths while minimizing individual weaknesses.

Implementing CrossBoost introduced complexities, such as managing heterogeneous model outputs, ensuring consistent feature scaling across learners, and optimizing memory usage to prevent EC2 compute bottlenecks. Additionally, training multiple boosting models increased processing time, requiring careful tuning of depth and estimator parameters. Despite these challenges, the ensemble's superior generalization showcases its robustness in detecting subtle anomalies within high-dimensional cloud workloads. The experiment confirms that ensemble learning significantly enhances detection reliability, making CrossBoost the most suitable model for operational cloud environments.

Table 3: Performance Comparison of Boosting Models Across Core Evaluation Metrics

Model	Accuracy	Macro Precision	Macro Sensitivity (Recall)	Macro F1
Gradient Boosting Classifier	0.91	0.91	0.91	0.91
Hist Gradient Boosting Classifier	0.88	0.89	0.88	0.88
XGB Classifier	0.92	0.92	0.92	0.92
LGBM Classifier	0.88	0.88	0.88	0.88
CatBoost Classifier	0.92	0.92	0.92	0.92
CrossBoost Meta Ensemble	0.95	0.95	0.95	0.95

6.4 Evaluation of Real-Time Inferencing Performance

The experiment of real-time inferencing ensured that the implemented system can track continuous streams of workloads, detect anomalies and issue notifications automatically. Workloads Simulated workloads on EC2 enabled testing of realistic patterns of cloud behaviour such as bursts and fluctuations in resources. The data processing loop using an inference loop processed data after every 10 seconds and was effective in identifying the possible failure indicators, which confirmed the alerting pipeline using SNS. This experiment shows that the proposed study goes further than offline modelling and implements operational anomaly detection and live monitoring that attempts to predict the behaviour and respond to values that are out of safe performance ranges.

6.5 Comparative Discussion with Baseline

My study demonstrates superior performance using advanced boosting classifiers and a novel CrossBoost meta-ensemble. While the baseline study given by (Xin et al., 2023) relies on unsupervised distance- and density-based methods with linear and deep ensemble integration, it still suffers from lower accuracy on noisy datasets and lacks strong feature selection or data balancing as shown in Table 4. In contrast, my study applies rigorous preprocessing (SMOTE, scaling, feature selection) and leverages powerful supervised boosting models that learn complex non-linear cloud patterns more effectively. As a result, my system achieves higher anomaly detection accuracy, stronger generalization, and clearer interpretability through LIME, making it more practical for real-world cloud environments.

Table 4: Comparative Analysis – Proposed Study vs. Baseline (Xin et al., 2023)

Aspect	Baseline (Xin et al., 2023)	My Study (CrossBoost + Boosting)
Learning Type	Unsupervised / Weakly Supervised	Fully Supervised Boosting
Data Processing	PCA only, no balancing	SMOTE + Scaling + Feature Selection
Models Used	IForest, KNN, LOF, OCSVM	XGBoost, LightGBM, CatBoost, GBM
Ensemble Type	Linear + Deep Ensemble	CrossBoost Meta-Ensemble
Accuracy	Moderate to good	Higher and more stable
Robustness	Varies across datasets	Strong generalization
Interpretability	Limited	LIME-based Explainability

7 Conclusion and Future Works

This study has designed a complete cloud anomaly detection system with the aid of the Google 2019 Cluster Trace, and considerable data cleaning, preprocessing, feature engineering, and model evaluation are implemented to address the variability of workloads of large scale. The approach combined outlier management, label coding, SMOTE balancing, scaling and top features with the aim of providing high quality input of classification. There are multiple boosting algorithms (GradientBoosting, HistGradientBoosting, XGBoost, LightGBM, and CatBoost) were tested and compared based on confusion matrices, classification metrics and latency, use of memory, and throughput.

The CrossBoost Meta-Ensemble constructed based on a stacking architecture as the meta-learner using LGBM showed the best predictive accuracy and robustness among all models. This proves the fact that heterogeneous boosting algorithms are highly effective in detecting anomalies than single models. The implementation on AWS (Cloud9, EC2, S3) also confirmed the scalability of the framework, which is appropriate to real-life cloud operation. Two methods to make AI explainable included explainable AI using LIME along with interpretable feature-level contributions to prediction of anomalies. This study also confirmed the implemented anomaly detection system on a live setting with synthetic workloads on the AWS EC2. Live inferencing showed that the model was capable of continuously tracking work patterns, to make a prediction, and perform SNS-based notification in case of an abnormal workload or failure risk was identified. This proves that the suggested solution is practical in a real working environment and can be further extended to similar automated monitoring processes as CloudWatch.

Future directions could include a way to incorporate deep learning models like LSTM, BiLSTM and Transformer based models to reflect more of the temporal behavior. This system could be enhanced with real-time streaming pipelines, SHAP-based global interpretability, cross-cloud evaluation, and automated remediation using reinforcement learning that would make it a fully autonomous and self-healing cloud infrastructure.

References

Ahmed, S.A., Khalifa, E.H., Nawaz, M., Abdalla, F.A., Mahmoud, A.F.A., 2025. Enhancing Cloud Data Center Security through Deep Learning: A Comparative Analysis of RNN, CNN, and LSTM Models for Anomaly and Intrusion Detection. Eng. Technol. Appl. Sci. Res. 15, 20071–20076. https://doi.org/10.48084/etasr.9445

Alghushairy, O., Alsini, R., Alhassan, Z., Alshdadi, A.A., Banjar, A., Yafoz, A., Ma, X., 2024. An Efficient Support Vector Machine Algorithm Based Network Outlier Detection System. IEEE Access 12, 24428–24441. https://doi.org/10.1109/ACCESS.2024.3364400

Al-Ghuwairi, A.-R., Sharrab, Y., Al-Fraihat, D., AlElaimat, M., Alsarhan, A., Algarni, A., 2023. Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. J. Cloud Comput. 12, 127. https://doi.org/10.1186/s13677-023-00491-x

Amiri, Z., Heidari, A., Navimipour, N.J., Unal, M., 2023. Resilient and dependability management in distributed environments: a systematic and comprehensive literature review. Clust. Comput. 26, 1565–1600. https://doi.org/10.1007/s10586-022-03738-5

Arunkumar, M., Kumar, K.A., 2023. GOSVM: Gannet optimization based support vector machine for malicious attack detection in cloud environment. Int. J. Inf. Technol. 15, 1653–1660. https://doi.org/10.1007/s41870-023-01192-z

George, A.S., A.S.Hovan George, T.Baskar, 2023. Edge Computing and the Future of Cloud Computing: A Survey of Industry Perspectives and Predictions. https://doi.org/10.5281/ZENODO.8020101

Girish, L., Rao, S.K.N., 2023. Anomaly detection in cloud environment using artificial intelligence techniques. Computing 105, 675–688. https://doi.org/10.1007/s00607-021-00941-x

Hagemann, T., Katsarou, K., 2020. A Systematic Review on Anomaly Detection for Cloud Computing Environments, in: 2020 3rd Artificial Intelligence and Cloud Computing Conference. Presented at the AICCC 2020: 2020 3rd Artificial Intelligence and Cloud Computing Conference, ACM, Kyoto Japan, pp. 83–96. https://doi.org/10.1145/3442536.3442550

Islam, M.S., Rakha, M.S., Pourmajidi, W., Sivaloganathan, J., Steinbacher, J., Miranskyy, A., 2025. Anomaly Detection in Large-Scale Cloud Systems: An Industry Case and Dataset, in: 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). Presented at the 2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), IEEE, Ottawa, ON, Canada, pp. 377–388. https://doi.org/10.1109/ICSE-SEIP66354.2025.00039

Jiang, J., Liu, F., Ng, W.W.Y., Tang, Q., Zhong, G., Tang, X., Wang, B., 2023. AERF: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing. Comput. Commun. 200, 86–94. https://doi.org/10.1016/j.comcom.2023.01.004

Keum, K., Kwak, J.Y., Rim, J., Byeon, D.H., Kim, I., Moon, J., Park, S.K., Kim, Y.-H., 2024. Dual-stream deep learning integrated multimodal sensors for complex stimulus

detection in intelligent sensory systems. Nano Energy 122, 109342. https://doi.org/10.1016/j.nanoen.2024.109342
Natha, S., Ahmed, F., Siraj, M., Lagari, M., Altamimi, M., Chandio, A.A., 2025. Deep BiLSTM Attention Model for Spatial and Temporal Anomaly Detection in Video Surveillance. Sensors 25, 251. https://doi.org/10.3390/s25010251
Oluwafemi Esan, 2024. ENHANCING SAAS RELIABILITY: REAL-TIME ANOMALY DETECTION SYSTEMS FOR PREVENTING OPERATIONAL DOWNTIME. https://doi.org/10.5281/ZENODO.15482517
S, A., D, S., G, P., 2023. Malicious insider threat detection using variation of sampling methods for anomaly detection in cloud environment. Comput. Electr. Eng. 105, 108519. https://doi.org/10.1016/j.compeleceng.2022.108519
Sahaaya Arul Mary, S.A., Anwar Basha, H., Mohanraj, G., Kiruthikaa, R., Saranya, N., 2025. Leveraging 5G and cloud computing for outlier detection in IoT environments: A KNN approach. Internet Technol. Lett. 8, e550. https://doi.org/10.1002/itl2.550
Sherubha, P., P. Sasirekha, S., Dinesh Kumar Anguraj, A., Vakula Rani, J., Anitha, R., Phani Praveen, S., Hariharan Krishnan, R., 2023. An Efficient Unsupervised Learning Approach for Detecting Anomaly in Cloud. Comput. Syst. Sci. Eng. 45, 149–166. https://doi.org/10.32604/csse.2023.024424
V., S., A., S., 2023. LSTM-Markov based efficient anomaly detection algorithm for IoT environment. Appl. Soft Comput. 136, 110054. https://doi.org/10.1016/j.asoc.2023.110054
Wang, Y., 2024. Network Anomaly Traffic Detection Using WGAN-CNN-BiLSTM in Big Data Cloud–Edge Collaborative Computing Environment. J. Inf. Process. Syst. 20, 375–390. https://doi.org/10.3745/JIPS.01.0105
Xin, R., Liu, H., Chen, P., Zhao, Z., 2023. Robust and accurate performance anomaly detection and prediction for cloud applications: a novel ensemble learning-based framework. J. Cloud Comput. 12, 7. https://doi.org/10.1186/s13677-022-00383-6
Zhang, H., Jia, X., Chen, C., 2025. Deep Learning-Based Real-Time Data Quality Assessment and Anomaly Detection for Large-Scale Distributed Data Streams. Int. J. Med. Body Health Res. 6, 01–11. https://doi.org/10.54660/IJMBHR.2025.6.1.01-11