National College of Ireland

Configuration Manual

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

Lecturer: 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: …1414…… **Page Count:** …5……

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):	

Configuration Manual

Suhail Kuzhikandathil Sageer
Student ID: 22248048

1. Introduction

The contemporary cloud platforms, including Google Borg, Kubernetes, and cluster management systems, handle millions of computing operations within a day. These systems should be reliable, fault tolerant and have good workload scheduling. Such environmental failures may cause service degradations, loss of jobs and downtimes, hence the importance of early failure detection.
The current project involves a data analysis of the Google Cluster Workload Trace 2019 that is a comprehensive data set of job submissions, scheduling behaviors, resource requests, resource usage, and system events that occur within 8 clusters of Google data centers. The data comprises CPU histograms, pattern of memory allocation, job-level associations, and execution properties.
The goal of the project is to forecast early job or node-level failures by estimating resource consumption pattern and job metadata via machine learning models - specifically boosting algorithms and a custom Stacking.

The project is especially applied to:

- Monitoring of the cloud infrastructure.
- Prediction of failure and Reliability Engineering.
- Optimization of Distributed Systems Scheduling.
- Anticipatory Maintenance within the Data Centers.
- AI-based Cloud Operations Research.

This project provides a scalable method of early warning of anomaly and failure of big cloud computing systems by integrating cluster resource metadata, temporal job behavior, and ensemble learning techniques.

2. Research Question

Primary Research Question

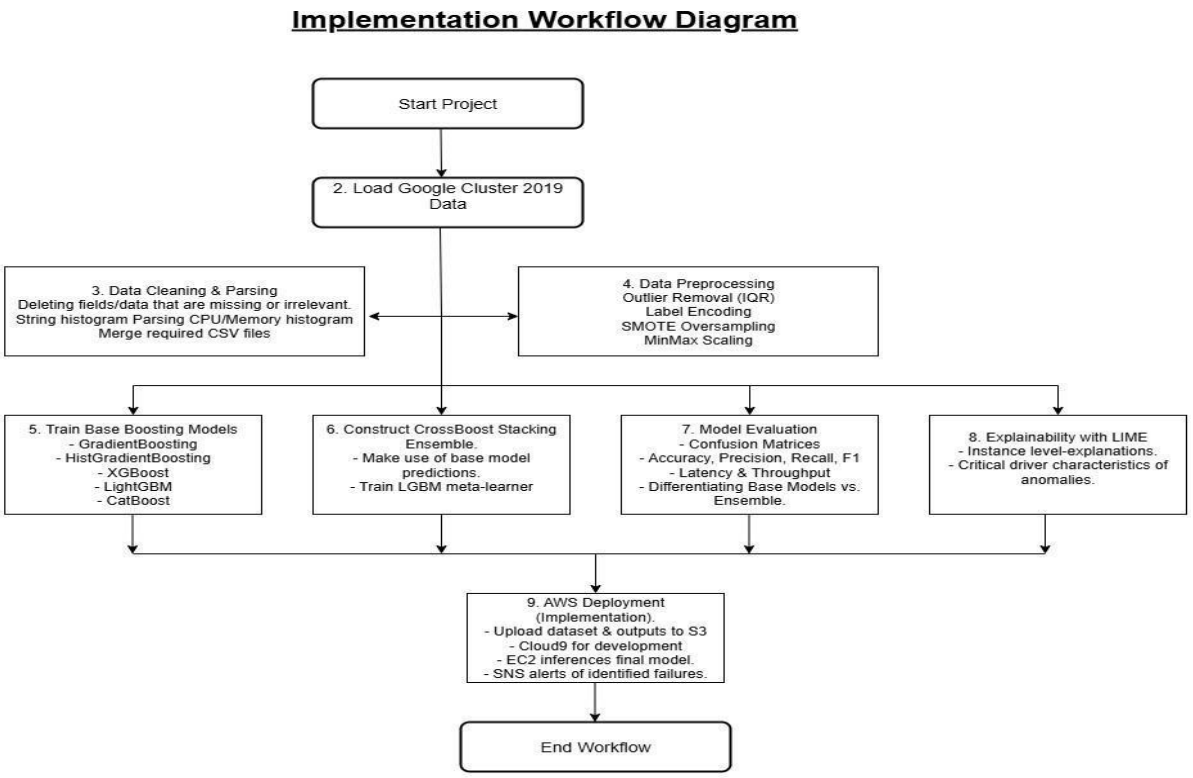
What combination of boosting techniques can most effectively detect cloud resource anomalies in large-scale cluster environments while ensuring high accuracy, low latency, and strong interpretability through Explainable AI?

2.1 Explanation of the Research Question

Cloud failures in the form of resource failures are caused by a combination of:

- CPU or memory spikes
- Sudden scheduling delays.
- Large change in resource requests and utilization.
- Job dependencies at several stages.
- Cluster-level contention

2.2 Project Implementation Workflow



The workflow diagram gives an overview of the implementation process we used in this project and thus how it started with loading the Google Cluster 2019 data up to the deployment of the model on AWS. It provides an overview of the key steps of data preparation, model training, ensemble creation, evaluation, and explainability and cloud deployment. The diagram is provided as a graphical representation to the general outline of the pipeline showing how raw cluster data is processed into clean and pre-processed data, boosting-based learning, ensemble prediction, performance evaluation, and ultimately deployed to run on operations with real-time alerting.

3. Environment Setup

Proper set up of the environment guarantees reproducibility of preprocessing, training models and deploying the models to AWS.

3.1 Recommended Hardware

Component	Recommended
CPU	Quad-core or higher
RAM	Minimum 16 GB (large dataset operations)
GPU	Optional (not required; models are CPU-based)
Storage	~10 GB for datasets & models

Boosting models are tuned to CPU environments and as such, there is minimal dependency on the GPUs.

3.2 Software Requirements

Supported operating systems:

- Windows 10/11
- Linux (should be used for long runs)
- macOS

3.3 Python Version

The notebook supports:

Python 3.8 – 3.11

3.4 Required Libraries

Core Libraries

- numpy
- pandas
- matplotlib
- seaborn

Machine Learning Libraries

- scikit-learn
- xgboost
- lightgbm
- catboost
- joblib

Explainable AI

- lime

System Utilities

- os
- time
- psutil
- warnings

AWS Services

- boto3 (for SNS alerting during failure detection)

3.5 Installation Commands

- pip install numpy pandas matplotlib seaborn
- pip install scikit-learn xgboost lightgbm catboost
- pip install lime joblib psutil boto3

4. Dataset Description

Original Data: Google Cluster Workload Trace 2019

The dataset is composed of job events, scheduling information, resource requests, usage statistics, histograms of CPU usage, and allow-set metadata in various Google clusters. It covers millions of records dividing in groups of various CSV files. Before analysis, they need to be cleaned and merged significantly.

4.1 Data Preprocessing

Processing involves the treatment of missing values, the elimination of irrelevant fields, the parsing of a column that is of the form of a JSON, and the transformation of datatypes. Outliers are addressed, categorical features are coded, imbalance is addressed, and features are scaled. The best 15 features chosen in Random Forest importance are used as the final training.

4.2 Exploratory Data Analysis

EDA looks at CPU and memory distributions, frequency of job states, correlation and trends in resource trends. Cluster-level plots indicate the fluctuation of workloads. Insights are utilized to determine workload trends and possible workload failure.

5. Model Architectures

Algorithms include GradientBoosting, HistGradientBoosting, XGBoost, LightGBM, CatBoost, and a stacking ensemble of crossboost. Base learners make predictions, which are amalgamated through an LGBM meta-classifier. The ensemble enhances strength by using heterogeneous boosting techniques.

5.1 Training Pipeline

Data is divided into 80/20, base models are being trained, and stacked meta-features are developed. LGBM meta-learner is then trained, and the final model is saved using joblib. The duration of training and utilization of resources are registered in performance assessment.

5.2 Evaluation Metrics

Accuracy, precision, recall, F1-score, specificity, and sensitivity are used to measure model performance. Latency, throughput, memory consumption and execution time are also tested. Classification reports and confusion matrices give some graphical information.

6. Explainable AI (LIME)

LIME can be used to provide an explanation of individual predictions in terms of the features that predict failure or non-failure outputs. Such explanations enhance transparency of the models. There are sample test cases that indicate the use of interpretations.

7. AWS Deployment & Real-Time Experiment

The model is implemented on AWS EC2 and Cloud9 to do real-time inference. Workload samples (synthetic) are processed after every 10 seconds, and predicted failures give out SNS email alerts. This certifies the operational preparedness to real-time cloud vigilance.

8. References

Ahmed, S. A., Khalifa, E. H., Nawaz, M., Abdalla, F. A. and 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, Engineering, Technology & Applied Science Research 15: 20071–20076.

Islam, M. S., Rakha, M. S., Pourmajidi, W., Sivaloganathan, J., Steinbacher, J. and Miranskyy, A. (2025). Anomaly detection in large-scale cloud systems: An industry case and dataset, ICSE-SEIP pp. 377–388.

Xin, R., Liu, H., Chen, P. and Zhao, Z. (2023). Robust and accurate performance anomaly detection and prediction for cloud applications: A novel ensemble learning-based framework, Journal of Cloud Computing 12: 7.

Girish, L. and Rao, S. K. N. (2023). Anomaly detection in cloud environment using artificial intelligence techniques, Computing 105: 675–688.

Al-Ghuwairi, A.-R., Sharrab, Y., Al-Fraihat, D., AlElaimat, M., Alsarhan, A. and Algarni, A. (2023). Intrusion detection in cloud computing based on time series anomalies utilizing machine learning, Journal of Cloud Computing 12: 127.

Wang, Y. (2024). Network anomaly traffic detection using WGAN-CNN-BiLSTM in big data cloud–edge collaborative computing environment, Journal of Information Processing Systems 20: 375–390.

Jiang, J., Liu, F., Ng, W. W. Y., Tang, Q., Zhong, G., Tang, X. and Wang, B. (2023). AERF: Adaptive ensemble random fuzzy algorithm for anomaly detection in cloud computing, Computer Communications 200: 86–94.