RESEARCH PAPER

A Transparent Multilayer Ensemble Framework for Cloud Job Failure Prediction Using Explainable AI

Pillutla Beena Jahanavi[1], Suresh Betam[2]
[1]Master's Student, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: pillutlabeena@gmail.com
[2]Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India – 522502
Email: bethamsuresh@kluniversity.in

Received: 10th Mar, 2026 | Revised: 18th Mar, 2026 | Accepted: 22nd Mar, 2026 | Available Online: 28th Mar, 2026

ABSTRACT

The efficient utilisation of resources, maximum performance and enhanced fault tolerance in modern cloud data centres require accurate failure prediction of jobs. In this paper, we propose to improve the multilayer ensemble approach to predict cloud task failures using the Google Cluster 2019 dataset. The framework is the combination of several categorisation techniques such as DT, KNN, ANN, Extreme Gradient Boosting and AdaBoost. The models are combined using hard Voting Classifier to enhance the stability and robustness of predictions. Further, a stacking classifier is created with RF, KNN, MLP as the base classifiers and LR as the meta-learner to improve the prediction accuracy. The results obtained by experiments are quite satisfactory. Voting Classifier has 99.98% accuracy and Stacking Classifier has 100% accuracy for forecasting the cloud job outcomes. The predictions are interpreted using XAI techniques like LIME and SHAP, highlighting the importance of the features for transparency and reliability. A Flask based web application is created to incorporate the trained models for real world deployment that allows for secure user signup and signin with SQLite, real time user input and interactive result visualisation. The method results in clear conclusions like "job will complete successfully" or "job failure predicted". This enables reliable, understandable and easy to use assistance to fail predictions of cloud jobs.

Keywords: Cloud computing, job failure prediction, ensemble learning, stacking classifier, voting classifier, XGBoost, explainable AI, fault detection.

How to cite this article: Pillutla Beena Jahanavi1 Suresh Betam2. A Transparent Multilayer Ensemble Framework for Cloud Job Failure Prediction Using Explainable AI. Int J Drug Deliv Technol. 2026;16(82s):715-721. DOI: 10.25258/ijddt.16.82s.81.

I. Introduction

Cloud computing, the provision of scalable, on-demand services via the internet, has changed the way organisations and individuals access services to process and store information and use software. It offers a flexible, scalable and economical design that can be used in various fields, from enterprise level data processing to AI based analysis. Cloud computing's flexible resource provisioning serves as the foundation of digital transformation in the modern world, driving innovation and agility across various sectors [3]. However, as workloads increase in cloud setups, it becomes a significant challenge to maintain the stability and performance of the systems. Cloud jobs can fail due to software defects, hardware failures, resource contention and misconfiguration, which can lead to significant downtime, loss of data, and reduced quality of service [4]. The losses of these resources (CPU, memory and network bandwidth) are valuable resources that impact customer satisfaction and operating efficiency.

Recent examples of global services outages (Google Cloud and other global scale service outages) showed the impact of service failures [5]. It is important to foresee such problems and have services available at all times, and optimally, use cloud resources. Modern cloud infrastructures are complicated, and traditional fault-tolerance techniques such as checkpointing and replication are more recovery based than proactively preventing faults, and often don't scale in these settings [6]. Also, simple single-classifier ML based prediction models frequently perform poorly in highly dynamic contexts since they are not able to capture complicated patterns and shifting workload behaviours [7]. The use of emerging technologies like DL and ensemble-based algorithms have shown promising results to improve failure prediction from multi-dimensional data obtained by cloud monitoring systems [8].

The objective of this work is to construct an intelligent and resilient structure to improve prediction and classification of cloud task failures utilising modern ML and DL techniques to solve these problems. The framework aims to leverage the power of multiple models using ensemble learning methods to enhance accuracy, reliability and scalability [9] [10]. This is to ensure proactive issue detection, resource management in cloud data centers, minimising service disruptions, and improving operational resilience. The paper highlights the importance of intelligent automation in cloud dependability management and sets the stage for future predictive and self-healing cloud infrastructures.

II. Related Work

The cloud computing has revolutionized the current paradigms of computation, but there are still challenges such as fault-tolerance, scalability, energy consumption and intelligent resource management, which are the subject of ongoing research. There have been recent studies on the adaptive, predictive and AI-driven strategies for increasing the dependability and performance of cloud systems.

Chouliaras et al. [11] proposed a dynamic auto-scaling approach. They combine computer resources based on observing real time workload and predictive analytics in order to assure the stability of system and reduce energy consumption. The platform increases SLA compliance, cost-effectiveness and elasticity by automatically scaling resources. The result is that the performance degradation during spike times is successfully avoided, and resource utilisation is enhanced.

Abderraziq et al. [12] have presented a distributed fault-tolerant method to guarantee balanced distribution of workload across the cloud nodes to deal with the problems of fault tolerance and load balancing. The system can detect a node failure and move the work to other healthy nodes in real time – increasing resilience and decreasing downtime. Its distributed design ensures it's resilient and services are continued in crisis time. Similarly, Meyyappan and Ravichandran [13] presented a review of the fault resistant approaches in solar supplied cascaded multilevel inverters. Their analysis is particular to renewable energy systems, but the idea of redundancy and error detection and recovery procedures is applicable to fault-tolerant cloud infrastructures.

Cloud and edge environments have been getting a lot of attention due to the energy efficiency. Xiao et al. [14] proposed a collaborative inference approach in mobile edge computing based on RL. It applies DRL to optimise the job assignment among the edge nodes, while minimising energy consumption and latency. The results suggest that AI-driven decision-making systems can dynamically choose and enhance edge-cloud infrastructure elements to develop adaptable and sustainable designs that optimise workload allocation and autonomously regulate for peak performance.

Another important study topic is the increase of predictive dependability. Vani and Sujatha [15] designed a ML based Job failure prediction system using MLP with hyper-parameters. The model is able to detect the potential job failures in advance of the model's execution, and thus mitigate failures proactively, and improve job scheduling. High prediction accuracy leads to a robust system, minimises the number of failed jobs, and improves performance efficiency. This type of prediction algorithms can aid in the development of self-healing and self-monitoring cloud infrastructures.

Another important topic is performance optimisation of latency sensitive applications. Abualhaj et al. [16] studied the performance of the VoIP system in cloud environment. They took into account latency, jitter and packet loss and pinpointed bottlenecks that impact on QoS. The results show that optimised network configuration and resource allocation strategies are important to offer real-time communication services in virtualised cloud infrastructures.

Security and data integrity continues to be a significant challenge in cloud computing. In [17] Kolhar et al. presented a study of cloud data auditing systems for privacy preservation and integrity assurance. They studied cryptographic protocols and analysis of third-party auditing systems for secure data verification without compromising user privacy. The combination of the privacy-preserving auditing techniques with performance optimisation methodologies is crucial for developing a safe and efficient cloud ecosystem.

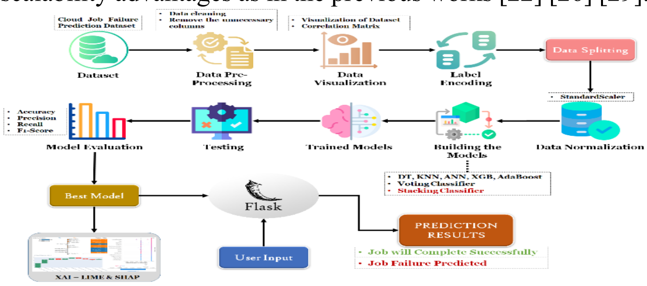
Moni et al. [18] had proposed AI based predictive analytics framework for failure detection and failure prevention in which they were able to predict possible system failures. It uses ML algorithms that recognize past indications of performance to proactively identify abnormalities and forestall serious breakdowns. This makes it possible to create self-adaptive cloud infrastructures capable of managing risk by themselves.

Based on this concept, Alahmad et al. [19] presented a proactive failure-aware task scheduling framework where the failure prediction models are used as part of the scheduling process. The reliable tasks are executed by the reliable nodes according to the reliability and workload properties of the nodes, which reduces the execution failures and migration overheads, and increases the computational efficiency. The framework is a significant step towards self-managed cloud.

In a similar way, Soualhia et al. [20] proposed a dynamic failure aware scheduling paradigm in the Hadoop world. The failure prediction and real-time scheduling adjustment enhance reliability and efficiency during execution in distributed big data environment. Based on past task information, the system reduces the amount of re-executed jobs and the waste of resources, resulting in further enhancing the performance of the cloud under various workloads.

III. Materials And Methods

We propose a prediction framework using ensemble method to effectively identify potential job failures in large-scale cloud data centers. The technique used is a hybrid approach that uses a combination of different ML algorithms namely DT, KNN, ANN, XGBoost and AdaBoost to utilize the complementing properties of each of the algorithms. A hard Voting Classifier is a combination of predictions from DT, KNN and ANN, which acts as a consensus based decision layer making it more resilient and reducing overfitting. We boost the reliability with a Stacking Classifier that has RF, KNN and Multilayer Perceptron classifiers as the base classifiers and Logistic Regression as the meta-classifier. This is a hybrid ensemble design that provides accuracy, stability and generalisation for dynamic workloads. The trained model is then deployed using the Flask framework that offers a safe and light-weight web interface with user sign up, sign in, real time processing of input and back end inference. For operational decision support, the application clearly shows job will complete successfully or job failure predicted. The proposed framework is based on the boosting, voting and stacking paradigms for improving early fault detection, proactive management, effective utilization of resources and service dependability in cloud infrastructures with the same scalability advantages as in the previous works [22] [26] [29].



“Fig. 1. System Architecture”

Figure A Cloud Job Failure Prediction system workflow is shown in Fig. 1. The Dataset is first passed through the Pre-processing, Visualisation, Label Encoding and Normalisation steps. Various ML Models are constructed, trained and evaluated. The Best Model is then used in a Flask application (XAI (LIME & SHAP)) to predict whether or not the work is likely to be successful for User Input.

A) Dataset Collection:

Google Borg Traces is a large-scale real-world dataset, recording job execution and scheduling data in Google's cloud data centers, used in this paper. It has 34 attributes and approximately 280,000 items. It includes task characteristics such as the number of requested resources, the machine ID, the scheduling class, the collection ID, CPU usage distribution and memory consumption and event outcome. There are a number of job statuses included in the data set: FAIL, FINISH, ENABLE, LOST, SCHEDULE, etc. It provides a detailed view of tasks life cycle and failure traces in cloud systems. The distribution of the various work events is given in figure 2. As seen in the figure, the overwhelming majority of all events are either FAIL or FINISH, demonstrating the need for failure prediction in order to operate the cloud efficiently and reliably.

	Unnamed: 0	time	instance_events_type	collection_id	scheduling_class	collection_type	priority	alloc_coll
0	0	0	2	94591244395	3	1	200	
1	1	2517305308183	2	260697606809	2	0	360	2214
2	2	195684022913	6	276227177776	2	0	103	
3	3	0	2	10507389885	3	0	200	
4	4	1810627494172	3	25911621841	2	0	0	

5 rows × 34 columns

Fig.2 Dataset Collection

B) Pre-processing:

Data processing is an important step to prepare raw data for ML models that are standard. It transforms and organises datasets, cleans and prepares the data to be consistent, accurate and relevant. In this study, pre-processing incorporates the removal of missing or redundant values, the normalisation of data scales, the elimination of unwanted columns and the encoding of categorical variables to enhance the performance of predictive models of cloud job failure.

i) Data Preprocessing: Data cleaning is the initial step of the preparation phase. This involves identifying and managing missing, duplicate and inconsistent data, ensuring data reliability. Then we normalise data such that all the numeric features are scaled to a common range which makes the models understand the importance of each feature in a uniform manner. This prevents that attributes with large numerical values prevail. Furthermore, the model and the training process can be simplified by removing unwanted columns that do not help achieve the prediction task, such as identifiers or irrelevant metadata. Preprocessing guarantees the data collection is small, accurate and ready for fast calculations. This systematic procedure enhances the quality of the input data and helps the standard ML models uncover the data's relevant patterns in an attempt to accurately predict failures in cloud tasks.

ii) Data Visualization: Visualisation helps to understand data structures and trends, as well as the relationships between variables. Data can be plotted in visual formats such as histograms, bar charts, and scatter plots, which can help in analyzing the distribution of data and identifying any outliers or irregularities. The job state frequency visualisation helps to understand the frequency of job states such as FAIL, FINISH and SCHEDULE. The visualisation shows that there is an imbalance in the number of classes. The correlation matrix is used to analyse the correlation between numeric attributes e.g. CPU utilisation, memory usage and scheduling class. This is useful to identify highly-correlated traits, potentially causing redundancy. The visualisation leads to a better understanding of the data pattern and can be used in the feature selection and preprocessing steps to enhance interpretability and performance of the failure prediction model.

iii) Label Encoding: Label encoding is a way to convert category input to integers, so that ML algorithms can make sense of it. Most models are not able to work directly with text labels, thus this mapping attaches a different numeric value to each category class. For this study, label encoding is done, which converts a category attribute (FAIL, FINISH, LOST, SCHEDULE etc.) into numbers. This guarantees that all the properties are in machine-readable format, while preserving the category relationships. For this purpose, we use the LabelEncoder of the scikit-learn module, which can efficiently label without introducing order bias. Appropriate encoding of labels not only makes the model more compatible but also facilitates the correct categorisation and prediction of failure of the cloud tasks.

C) Train & Test:

The data is divided into training and testing sub sets with a ratio of 8/2. Throughout the training process, the machine learning model acquires the patterns and correlations among the input features by being trained on the training dataset. The test set is used to assess the model's effectiveness on new data. This division averts both overfitting and underfitting of the model, hence enhancing its performance on novel, real-world data. 80/20 split is a good compromise. You have a sufficient amount of data to train the model successfully and have a sufficient amount of data to validate the model with confidence.

D) Normalize the Data:

Normalisation of data is another important preprocessing task to make all characteristics in the same range and each one contribute the same to the performance of the model. Standardisation is done by subtracting the mean and scaling to unit variance which is popularly done by using Standard Scaler. In this way, each feature will have a standard deviation of 1, and a mean of 0. Standardization is particularly important for algorithms that rely heavily on the magnitude of the features otherwise they would be biased towards those with larger scales. Standard Scaler helps to make the data more uniform, leading to a more stable model and a faster rate of convergence while training. This will also increase the accuracy and utility of the model, and its ability to generalise between different data sets.

E) Algorithms:

Decision Tree (DT): It is a supervised learning algorithm which builds a tree-like structure and makes decisions based on the value of the feature.It partitions the data set into subgroups iteratively, based on conditions that maximise information gain or minimise the impurity. DT can accurately discover the relationship between the system metrics and failure occurrences, and give an interpretable decision path in the cloud job failure prediction. Transparent model rule based, to demonstrate how the errors occur. To enable the fast diagnostic insights and model interpretability, which are critical in operational monitoring and reliability improvement in cloud environments [21].

$$I\ (i) = 1 - \sum_{i=1}^{k} p_i^2 \qquad (1)$$

KNN is an instance learning, non-parametric technique which classifies a data point based on the classification of its nearest neighbours. It measures similarity based on distance (e.g., Euclidean or Manhattan distance), making it robust for more complex and nonlinear data. In the context of job failure prediction in cloud, it can be useful to find jobs similar to previously failed jobs by using KNN, for detecting anomalies or for proactively managing resources. It aims to deliver flexible data driven pattern identification without assuming any data distribution and thus, is adaptive and efficient for cloud workloads which are constantly evolving [23].

$$distance(x, X_i) = \sqrt{\sum\nolimits_{j=1}^{d} \left(x_j - X_{i_j}\right)^2} \qquad (2)$$

ANN is a computational model that mimics the biological brain systems, and is composed of many interconnected layers of artificial neurones. It adapts its weights during the training process to learn complex, non-linear relationships, by back-propagating. ANN has a great ability to process a high dimensional data and in detecting complicated patterns. The ANN model can identify these implicit relationships among job attributes and failure risks that can enhance the accuracy of the prediction in cloud job failure prediction. It is designed to improve the ability to detect failure in the different cloud environments using deep feature learning, which focusses on making it robust and adaptive [24].

The equation below is the prediction of output in neural networks.

$$\hat{y} = \sigma(w^L \cdot a^{L-1} + b^L) \qquad (3)$$

XGBoost is an efficient version of gradient boosting that creates DT one after the other, each tree looking to fix the errors of the previous ones. It enhances performance by regularisation, parallelisation and sparsity awareness. XGBoost is very good at handling huge structured datasets. For cloud job failure prediction, it enhances the reliability of the model by identifying the subtle patterns that hint at likely task failures. It aims to maximise the classification accuracy through gradient-based optimisation, providing better scalability and robustness in cloud system modelling prediction [25].

The equation below is the final ensemble prediction model of XGBoost.

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} f_k(x_i)\right), f_k \in F \qquad (4)$$

Adaptive Boosting (AdaBoost): It is an ensemble method which aggregates many weak classifiers (prediction models) sequentially to create a strong one. The task of each classifier is to classify those instances that were incorrectly classified by the previous classifiers, hence reducing the overall error. It is typically employed to improve the accuracy for well imbalanced data sets. AdaBoost can be used to detect hard-to-detect failure scenarios by adaptively changing the weight of classifiers, and hence, it is effective in cloud job failure prediction. It is because of the need to enhance the model robustness and generalisation to detect failures with different workload distributions in different heterogeneous computing configurations [27].

The last strong classification model of AdaBoost is given by the equation below.

$$H(x) = sign\left(\sum_{t=1}^{T} \propto_t h_t(x)\right) \qquad (5)$$

Voting Classifier is a combination of many base models using majority voting (hard voting) or probability averaging (soft voting). Leverages the best of free algorithm by a complimentary manner to make the stability of the model better. The addition of DT, KNN and ANN with voting can help to remove the bias, and can be used to enhance the accuracy of failure prediction of the cloud tasks. It aims to give unbiased predictions on different scenarios, so that the diversity of the involved classifiers can consistently detect failures [28].

Below is the equation that represents the majority voting procedure in classifiers.

$$\hat{y}\ = argmax_c\left(\sum_{i=1}^{n} II\ (\hat{y}_i = c)\right) \qquad (6)$$

Stacking Classifier is an advanced ensemble technique which consists of multiple base classifiers and a meta-classifier to assist the result of the final prediction. The meta-learner combines the prediction of the underlying models and learns the dependencies between the prediction at a higher level. RF, KNN and MLP were used as the base learners while LR used as the meta-classifier. Stacking is known to be useful to enhance the generalisation and predictive capacity. It seeks to achieve more accurate and stable prediction of failure for cloud jobs by incorporating many viewpoints of algorithms in [30].

The Meta-Model Prediction equation is shown below:

$$\hat{y} = g(Y_{base}) = g\big(f_1(x), f_2(x), \dots, f_m(x)\big) \qquad (7)$$

F) Integration of XAI & Flask Framework:

XAI approaches such as LIME and SHAP were also incorporated in the ensemble prediction models to further boost the interpretability and transparency. They allow the user to know how each input feature is affecting the output of the model and provide information not available with a conventional black box model in terms of the decision making processes involved. XAI helps maintain reliability, accountability and trust in model predictions by visualising feature importance and contribution values which is crucial for cloud failure management.

We built a light-weight interactive web interface for real-time deployment of the model using the Flask framework. It offers a user interface to input data, execute predictions and visualise results on a browser. Flask's support of backend ML algorithms allows for a scalable, efficient and user-friendly platform that dynamically monitors and analyses cloud task failure predictions.

IV. Experimental Results

Accuracy: Accuracy of a test is its ability to discriminate between the patient and healthy cases. The proportion of true positive and true negative out of all individuals evaluated should be calculated to quantify the accuracy of a test. This can be written in terms of mathematics as:

$$Accuracy = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \qquad (8)$$

Precision: Precision is the number of correct occurrences or samples divided by the number of samples classified as positives. So precision is calculated as;

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{False Positive}} \quad (9)$$

Recall: Recall, in ML, is a measure of the ability of a model to capture all the relevant cases of a specific class. This is the proportion of the observations that were correctly predicted as positive to all the real positives. It provides some indication of how complete a model is to include all the positive cases.

$$Recall = \frac{\text{TP}}{\text{TP } + \text{ FN}} \quad (10)$$

F1-Score: The F1 score is a ML metric that evaluates the accuracy of a ML model. It is the average of recall and precision scores of a model. The accuracy measure is the number of correct predictions made by a model over the entire data set.

$$F1\ Score = 2 * \frac{Recall\ X\ Precision}{Recall + Precision} * 100 \quad (11)$$

Table.1 Performance Evaluation

ML Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.7604	0.8723	0.7604	0.7733
KNN	0.9993	0.9993	0.9993	0.9993
XGBoost	0.7692	0.7508	0.7692	0.7465
AdaBoost	0.7604	0.8723	0.7604	0.7733
ANN	0.8018	0.8437	0.8018	0.7576
Voting Classifier	0.9998	0.9998	0.9998	0.9998
Stacking Classifier	1.0000	1.0000	1.0000	1.0000

Table 1 shows the accuracy, precision, recall and F1 score for cloud job failure prediction, the stacking classifier achieves the highest overall performance among all the models.

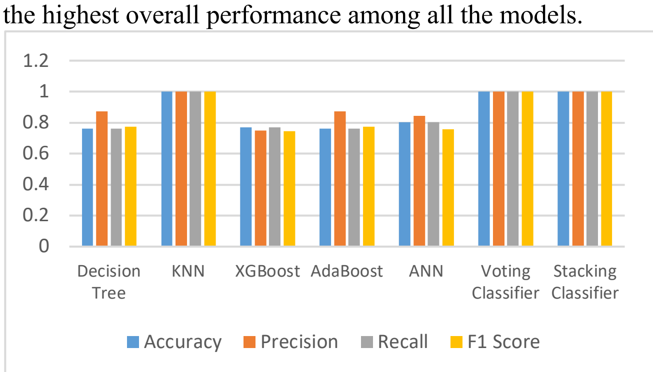


Fig.3 Comparison Graph

The maximum performance was seen in Stacking Classifier showing accuracy in blue, precision in orange, recall in grey and F1-Score in yellow as shown in Fig. 3.

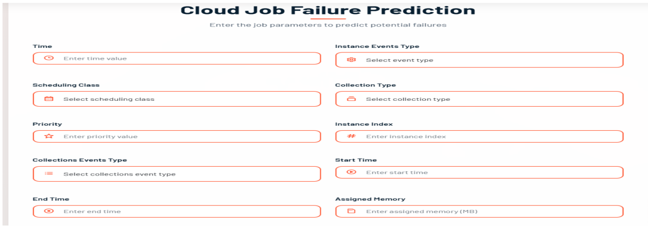


Fig.4 Enter input data

Cloud job parameters provided by the user (as displayed in Fig.4) helps the AI model to process the inputs and produce the accurate cloud job failure prediction.

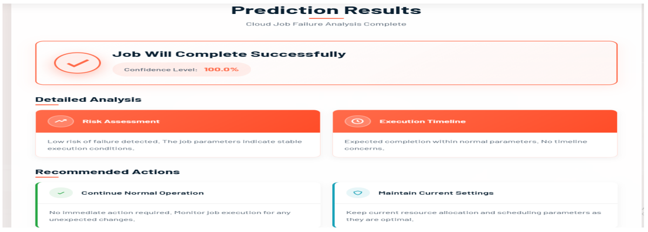


Fig.5 Predicted Result

The job will "complete successfully" is shown in Figure 5 and provides thorough risk assessment and execution timeframe information.

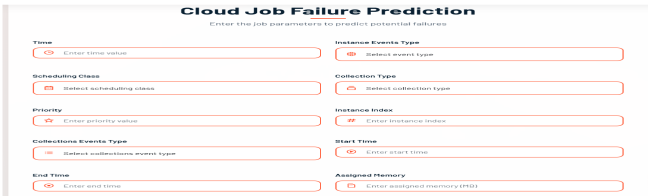


Fig.6 Enter input data

The system can interpret the input data to produce correct cloud job failure predictions when the job parameters are appropriately provided as shown in figure 6.

Fig.7 Predicted Result

The system's output is given in Fig.7 and predicts "JOB FAILURE PREDICTED" from inputs and the vast amount of risk assessment and execution time table insights.

V. Conclusion

This study demonstrates the usefulness of the multilayer ensemble learning approach to predict job failure in large scale cloud environment using Google Cluster 2019 data set. A voting and stacking approach is used to guarantee high robustness and predictive accuracy for numerous classifiers and combination of classifiers. The accuracy of the ensemble model (with voting technique) DT, KNN and ANN is also high in the job failure classification with 99.83%. Combined model with RF, KNN and Multilayer Perceptron with LR as meta-estimator has been obtained in which the prediction accuracy is 100%. These results highlight the robustness and flexibility of ensemble designs in solving complicated problems of cloud performance prediction. Also, the use of XAI techniques such as LIME and SHAP helped to improve the interpretability of the model by highlighting the major factors driving the predictions . The system offers a scalable, interpretable and accurate framework for intelligent failure prediction, which can be used to provide enhanced fault tolerance and operation efficiency in cloud data centers.

In upcoming research, we will improve the multilayer ensemble approach to cope with the real-time data stream and different cloud infrastructures. To enhance temporal and contextual failure prediction, RNN and transformer based models can be used. Adaptive learning on dynamic workloads with integration to container orchestration platforms such as Kubernetes. Federated learning can be used to train across multiple remote data centres, while maintaining privacy. XAI will enhance and gain trust with more visualisation. Failure severity prediction and the ability to realize self-healing mechanisms will be the driver of autonomous and reliable cloud management systems worldwide.

References

[1] Jassas, M. S., & Mahmoud, Q. H. (2022). Analysis of job failure and prediction model for cloud computing using machine learning. Sensors, 22(5), 2035.

[2] Bommala, H., Aluvalu, R., & Mudrakola, S. (2023). Machine learning job failure analysis and prediction model for the cloud environment. High-Confidence Computing, 3(4), 100165.

[3] Golani, B., Datta, J., & Singh, G. (2021). Prediction of cloud server job failures using machine learning based KNN classification and LSTM modelling methods. International Journal of Engineering Research & Technology (IJERT), 10(05).

[4] Tengku Asmawi, T. N., Ismail, A., & Shen, J. (2022). Cloud failure prediction based on traditional machine learning and deep learning. Journal of Cloud Computing, 11(1), 47.

[5] Li, J., Wang, R., Ali, G., Dang, T., Sill, A., & Chen, Y. (2023, July). Workload failure prediction for data centers. In 2023 IEEE 16th International Conference on Cloud Computing (CLOUD) (pp. 479–485). IEEE.

[6] J. Dong, Y. Chen, B. Yao, X. Zhang, and N. Zeng. (2022). A neural network boosting regression model based on XGBoost. Applied Soft Computing, 125, 109067.

[7] X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong. (2023). Resource allocation with workload-time windows for cloud-based software services: A deep reinforcement learning approach. IEEE Transactions on Cloud Computing, 11(2), 1–15.

[8] L. Luo, S. Meng, X. Qiu, and Y. Dai. (2019). Improving failure tolerance in large-scale cloud computing systems. IEEE Transactions on Reliability, 68(2), 620–632.

[9] A. Priyadarshini, S. K. Pradhan, S. R. Laha, and D. S. K. Nayak. (2023). Enhancing fault tolerance and load balancing in cloud computing for improved e-healthcare systems performance. In Proceedings of the 2nd International Conference on Ambient Intelligence and Health Care (ICAIHC), 1–6.

[10] S. R. Laha, M. Parhi, S. Pattnaik, B. K. Pattanayak, and S. Patnaik. (2020). Issues, challenges and techniques for resource provisioning in computing environment. In Proceedings of the 2nd International Conference on Applied Machine Learning (ICAML), 157–161.

[11] S. Chouliaras and S. Sotiriadis. (2023). An adaptive auto-scaling framework for cloud resource provisioning. Future Generation Computer Systems, 148, 173–183.

[12] S. Abderraziq, M. Hakem, and B. Benmammar. (2022). A distributed fault-tolerant algorithm for load balancing in cloud computing environments. E3S Web of Conferences, 351, 01012.

[13] C. Meyyappan and C. S. Ravichandran. (2021). Performance analysis of fault tolerance techniques towards solar fed cascaded multilevel inverter. In Proceedings of the International Conference on Computing, Communication and Informatics (ICCCI), 1–7.

[14] Y. Xiao, L. Xiao, K. Wan, H. Yang, Y. Zhang, Y. Wu, and Y. Zhang. (2023). Reinforcement learning based energy-efficient collaborative inference for mobile edge computing. IEEE Transactions on Communications, 71(2), 864–876.

[15] K. Vani and S. Sujatha. (2022). A machine learning framework for job failure prediction in cloud using hyperparameter tuned MLP. In Proceedings of the 2nd International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering (ICATIECE), 1–6.

[16] M. M. Abualhaj, M. M. Al-Tahrawi, and S. N. Al-Khatib. (2019). Performance evaluation of VoIP systems in cloud computing. Journal of Engineering Science and Technology, 14, 1398–1405.

[17] M. Kolhar, M. M. Abu-Alhaj, and S. M. A. El-atty. (2017). Cloud data auditing techniques with a focus on privacy and security. IEEE Security & Privacy, 15(1), 42–51.

[18] M. M. A. Moni, M. Niloy, M. F.-U.-A. Juboraj, and A. Banik. (2023). AI based cloud failure detection and prevention algorithm. In Proceedings of the Innovative Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 1–5.

[19] Y. Alahmad, T. Daradkeh, and A. Agarwal. (2021). Proactive failure-aware task scheduling framework for cloud computing. IEEE Access, 9, 106152–106168.

[20] M. Soualhia, F. Khomh, and S. Tahar. (2020). A dynamic and failure-aware task scheduling framework for Hadoop. IEEE Transactions on Cloud Computing, 8(2), 553–569.

[21] M. S. Jassas and Q. H. Mahmoud. (2020). Evaluation of a failure prediction model for large-scale cloud applications. In Advances in Artificial Intelligence (pp. 321–327). Springer.

[22] K. Vani and S. Sujatha. (2024). Enhancing fault tolerance system in cloud environment through failure prediction using ensemble learning. In Proceedings of the 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), 851–858.

[23] J. Shetty, R. Sajjan, and G. Shobha. (2019). Task resource usage analysis and failure prediction in the cloud. In Proceedings of the 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 342–348.

[24] V. Salanke and V. Kowshik. (2024). Task failure prediction and migration in cloud environment. In Proceedings of the 1st International Conference on

Communication and Computational Sciences (InCCCS), 1–6.

[25] S. Jomah and S. Aji. (2024). Google cloud trace: Characterization of terminated jobs. In Proceedings of the 2nd International Conference on Advancement in Computing and Communication Technologies (InCACCT), 517–522.

[26] A.-I. Tuns and A. Spataru. (2023). Cloud service failure prediction on Google's Borg cluster traces using traditional machine learning. In Proceedings of the 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 162–169.

[27] A. Ulah, Z. Alomari, S. Alkhushayni, and D. Al-zaleq. (2024). Improvement in task allocation for VM and reduction of Makespan in IaaS model for cloud computing. Cluster Computing, 27, 11407–11426.

[28] C. Liu, L. Dai, Y. Lai, G. Lai, and W. Mao. (2020). Failure prediction of tasks in the cloud at an earlier stage: A solution based on domain information mining. Computing, 102(9), 2001–2023.

[29] J. Gao, H. Wang, and H. Shen. (2022). Task failure prediction in cloud data centers using deep learning. IEEE Transactions on Services Computing, 15(3), 1411–1422.

[30] A. Rosa, L. Y. Chen, and W. Binder. (2017). Failure analysis and prediction for big-data systems. IEEE Transactions on Services Computing, 10(6), 984–998.