IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

A Multilayer Ensemble Learning Framework for Cloud Job Failure Prediction

Vibha Vishwakarma[1], Ms. Dolly Verma[2]

Student, Department of Computer Science and Engineering, RSR Rungta College of Engineering and Technology, Bhilai Chhattisgarh 490001[1]

Assistant Professor, Department of Computer Science and Engineering, RSR Rungta College of Engineering and Technology, Bhilai Chhattisgarh 490001[2]

Abstract: To achieve high performance, efficient resource utilization, and better fault tolerance, it is important for cloud data centers to predict job failures accurately; it is essential for the cloud data center to correctly predict job failures. This paper proposes a novel multilayer ensemble-based approach for cloud job failure prediction using the Google Cluster 2019 dataset. The framework uses several classification algorithms, including Decision Tree (DT), K-Nearest Neighbors (KNN), Artificial Neural Network (ANN), Extreme Gradient Boosting (XGBoost), and AdaBoost. These models are combined using a hard Voting Classifier to increase the prediction stability and robustness of the prediction. Furthermore, a Stacking Classifier is constructed with the base learners: Random Forest, KNN, and MLP, and the meta estimator: LR to improve predictive accuracy further. The experimental results show strong predictive performance. The Voting Classifier achieved 97.83% accuracy, while the Stacking Classifier achieved 97.98% accuracy in predicting cloud job outcomes. The use of explainable AI methods, such as LIME and SHAP, ensures transparency and reliability, and helps to explain the prediction and promote the value of the features. The trained models are integrated into a Flask-based web application that supports user sign-up and sign-in through SQLite, along with real-time processing and presentation of user inputs. The system provides an understandable judgement of "job will successfully complete" or "job failure predicted" for reliable, interpretable and userfriendly cloud job failure prediction.

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

I. INTRODUCTION

Cloud computing has brought a major shift in how organizations and users access computing power, storage, and software resources through the Internet on an on-demand and scalable basis. It provides a flexible, cost-effective, and efficient architecture for many applications, from enterprise-level data processing to AI-driven analytics [1], [2]. Cloud computing, the key technology behind the ongoing digital revolution, allows resources to be allocated dynamically and promotes innovation and agility in all sectors [3]. However, with increasing workloads in cloud settings, it has become a huge problem to maintain reliable and good performing systems. Failures in cloud jobs may be caused by software flaws, hardware breakdowns, resource contention, or configuration problems, which may lead to major downtime, data loss and decreased quality of service [4]. Such failures waste resources such as CPU, memory and network bandwidth, which can negatively affect user experience and operational efficiency.

System failures affect many systems, with recent examples of Google Cloud and other large-scale services failing [5]. It is crucial to be able to predict these failures in advance to maintain the availability of services and to maximize the use of cloud resources. However, many existing fault-tolerant techniques are mainly recovery-oriented and may face limitations when applied to complex and dynamic cloud environments [6]. In addition, simple ML based prediction models based on single classifiers usually have poor performance in highly dynamic contexts since such models cannot capture complex patterns and develop workload characteristics [7]. New approaches, such as Deep Learning (DL) and ensemble-based algorithms, have shown promising results in predicting failures from multidimensional data collected by cloud monitoring systems [8].

To tackle such challenges, the current research aims to build an intelligent and resilient system for better prediction and classification of cloud task failures using advanced ML and DL techniques. The system uses ensemble learning approaches [9], [10] to combine the strengths of different models and improve prediction accuracy, reliability, and scalability. The aim is to detect the problems proactively in the cloud data centre and manage the resource appropriately to minimise service disruptions and to enhance operational resilience. The work not only underscores the importance of

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

intelligent automation in the management of cloud dependability but also points towards the foresighted and self-repaired cloud landscapes of tomorrow.

II. RELATED WORK

Although cloud computing has introduced new paradigms of modern computation, problems such as fault tolerance, scalability, energy efficiency and smart resource allocation are still subjects of research. In recent times, solutions that utilise the concept of adapting to cloud conditions, predicting cloud conditions, and the use of AI have been suggested to enhance cloud reliability and performance.

Chouliaras and Sotiriadis [11] suggested an adaptive auto-scaling solution for dynamic resource provisioning in the cloud. They dynamically allocate computational resources based on the current demand, and monitor and predict the demand, ensuring energy efficiency and stability of the system. It automatically scales resources to meet SLA compliance, cost effectiveness, and elasticity. The approach is effective in mitigating degradation in performance under workload surges and boosts overall resource utilization.

Abderraziq et al. [12] presented a distributed fault-tolerant method to guarantee load balancing between cloud nodes with regard to load balancing and fault tolerance. The algorithm quickly detects failure of nodes and transfers tasks to healthy nodes for real-time performance, thus improving the system's resilience and reducing downtime. Its distributed architecture enhances reliability and allows the provision of continuous services in adverse conditions. Meyyappan and Ravichandran [13] also studied fault-tolerant techniques of solar energy based cascaded multilevel inverter. Though the study focuses on the renewable energy systems, the concepts discussed with respect to redundancy and error detection and recovery mechanisms are conceptually relevant to the fault tolerant cloud infrastructures.

The cloud and edge environments have gained significance in terms of energy efficiency. Xiao et al. [14] suggested a reinforcement learning based collaborative inference model for mobile edge computing. The model uses DRL to improve work allocation among edge nodes for minimum energy consumption and latency. According to their results, AI-driven decision-making can result in adaptive and sustainable edge-cloud architectures that can optimise for itself and intelligently manage workloads.

It is also demonstrated that the improvement of predictive dependability is another important study area. Vani and Sujatha [15] introduced job failure prediction system using ML with hyperparameter controlled MLP. The model can anticipate the potential work failures ahead of execution, enabling proactive mitigation and improved task scheduling. The accuracy of predictions increases the robustness of the system, reduces unsuccessful jobs and enhances the efficiency of the system. These prediction algorithms help to create self-healing and self-monitoring cloud systems.

We also work on performance optimization of latency-sensitive applications. Abualhaj et al. [16] studied the performance of VoIP systems in Cloud environments. They examined ways to delay, jitter and drop packets to identify QoS bottlenecks. The findings highlight the need for better network configurations and effective resource provisioning techniques to ensure reliable real-time communication in virtualized cloud environments.

Security is still a huge concern for cloud computing, and so is data integrity. The privacy preservation and integrity assurance have been the focus of cloud data auditing systems studied by Kolhar et al. [17]. They worked on cryptographic protocols and third-party auditing frameworks for secure data verification while preserving user privacy. Incorporating privacy preserving auditing tools along with performance optimisation strategies is one of the crucial parts to build secure and efficient cloud ecosystems.

The concept of an AI-based predictive analytics system was proposed by Moni et al. [18] to detect potential faults in the system, which is a key component in failure detection and prevention. The system proactively detects the abnormalities and prevents serious failures by ML models that are based on past performance measurements. It is possible to design cloud infrastructures that can self-regulate risk.

Alahhad et al. [19] suggested a proactive failure-aware task scheduling system, which incorporates failure prediction models in the task scheduling selection process. Tasks are mapped onto reliable nodes based on their expected reliability and workload conditions to minimise the likelihood of failures in task execution and migration time to reliable nodes, and increase computational efficiency. This is a giant stride toward self-administered cloud services.

Similarly, Soualhia et al. [20] proposed a dynamic failure-aware scheduling approach to Hadoop ecosystems. The proposed failure prediction technique is integrated to the real-time scheduling algorithm to enhance reliability and execution efficiency of the distributed large data settings. This framework will use task history to avoid re-execution time and waste of resources for jobs. It also improves the performance of the cloud when delivering certain workloads.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

III. MATERIALS AND METHODS

In this paper, the proposed approach outlines an ensemble-based predictive model that can be used to precisely identify likely failure of jobs in the cloud. The technique uses complimentary characteristics and is based on a combination of several ML algorithms, such as DT, KNN, ANN, XGBoost and AdaBoost. A hard Voting Classifier is a combination of the predictions of DT, KNN and ANN to make a consensus-based decision layer which enhances resilience and lowers overfitting. The base learners are RF, KNN, MLP, and the meta estimator is LR to make it more reliable. This is a hybrid ensemble design that is optimal for accuracy, stability, and generalization with dynamic workloads. The trained model is deployed using the Flask framework, which is a lightweight web framework with a secure web interface, user signup, sign-in, real time input processing, and backend inference. The application generates clear "job will complete successfully" or "job failure predicted" with operational decisions so they can be made. The framework introduces the boosting, voting and stacking paradigms as described in [22], [26] and [29] and scalability benefits cloud infrastructures for early fault detection, proactive management, effective resource usage and service dependability.

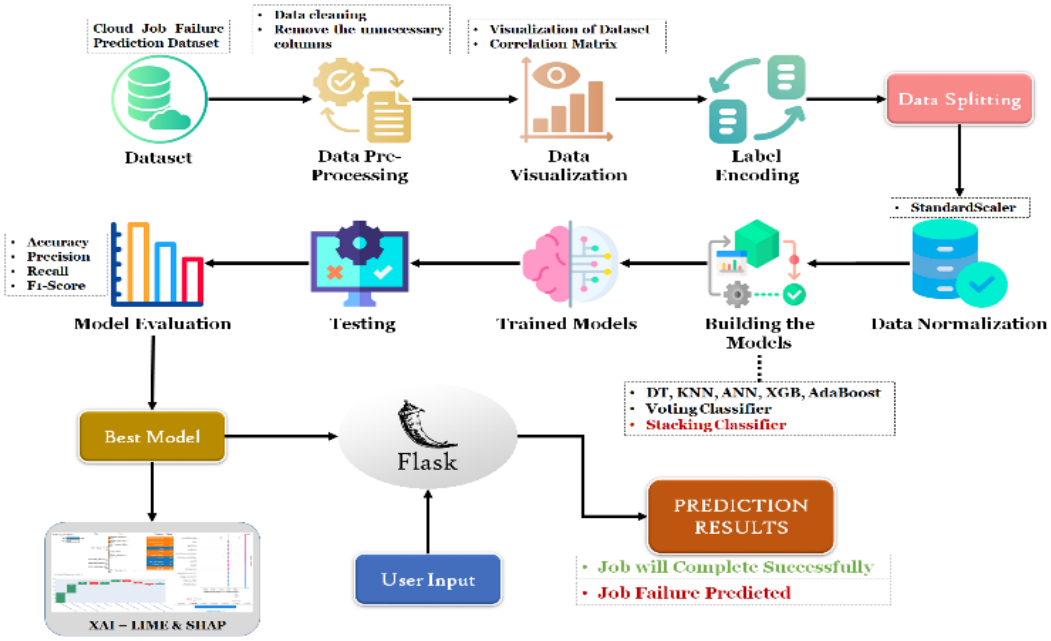


Figure. 1. System Architecture

The Cloud Job Failure Prediction system is a workflow as shown in Figure 1. The Dataset goes through Pre-processing, Visualization, Label Encoding and Normalization. There are numerous ways to develop, train and evaluate ML Models. Then the Best Model is integrated into a Flask application along with XAI (LIME & SHAP) to predict whether a job will be successful or not based on User Input.

A) Dataset Collection:

For this research, Google Borg Traces . is used. It is a large-scale real-world dataset that has job execution and schedule information from Google's cloud data centers. It holds over 280,000 entities and 34 attributes which describe the attributes of the task itself, such as resources needed, machine ID, scheduling class, collection ID, distribution of CPU usage, memory usage, and the result of the events of the task. The dataset includes various states of jobs like FAIL, FINISH, ENABLE, LOST, SCHEDULE etc that provide a comprehensive view of the job life cycle and failure behaviors in the cloud environment. The types of job events are distributed as in Figure 2. Most of the events are FAIL and FINISH which suggests failure prediction is essential for efficient and reliable operations of cloud systems.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

	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

Figure.2 Dataset Collection

B) Pre-processing:

Data preprocessing is crucial for obtaining data suitable for ML models. It cleans, transforms, and organizes datasets to guarantee consistency, accuracy, and relevance of data. In this study, the data is preprocessed by handling missing and duplicate values, normalizing the data, removing unnecessary columns, and converting categorical values into a suitable format to improve the performance of the cloud job failure prediction models.

i) Data Preprocessing: The first step is the data cleansing process, which identifies missing, duplicate and inconsistent records, and takes steps to correct them and ensure the integrity of the data. After that, data normalization is done, so that all numbers in a range between 0 and 1. This enables the models to provide a consistent understanding of the feature of importance. This has the effect of not allowing those qualities which have a big number to dominate. Furthermore, irrelevant columns (such as identifiers or other metadata that is not relevant to the prediction task) are removed from the data to further reduce the complexity and time required for the model to be trained. Pre-processing ensures small, accurate, and efficient computing data sets. The methodical technique enhances the quality of input data, making ML models capable of discovering meaningful patterns for accurate prediction of cloud task failures.

ii) Data Visualization: Data visualization enables us to comprehend the framework of a data set, patterns, and connections between variables. Then we look at the data distribution and try to detect any possible outliers / anomalies by using visuals such as histogram, bar plot, scatter plot etc. looking at the data pattern provide understanding of frequency of a job state such as FAIL, FINISH, SCHEDULE etc. This shows the imbalance in the distribution of classes. The correlation matrix is used to analyze the correlation between the numerical parameters such as CPU utilization, memory usage, scheduling class, etc. This is helpful to identify strongly linked traits, which can result in redundancy. By using visualization techniques, the patterns in the data can be better understood, which can inform feature selection and data preprocessing steps, enhancing the interpretability of the failure prediction model and its performance.

iii) Label Encoding: Label encoding is used to convert our category data into a numerical format ML algorithm can use. This is a mapping of each categorical class to an integer value, as many models don't support text labels directly. In this study, label encoding is used to convert categorical attributes such as job event categories (e.g., FAIL, FINISH, LOST, SCHEDULE) into numbers. This guarantees that all the properties are machine readable while preserving the category relationships. It is done using LabelEncoder from the scikit-learn module that is efficient in encoding and doesn't introduce the ordinal bias. Properly encoding labels with suitable encoding methods helps to improve compatibility with the models, as well as appropriate classification and prediction of the failure of cloud jobs.

C) Train & Test:

The dataset was divided into 80% training data and 20% testing data. The training set was used to learn patterns from the input features, while the testing set was used to evaluate model performance on unseen data. This separation helps assess how well the trained models generalize new observations. The training set is used to train the ML model and allow it to discover patterns and relationships between the characteristics (features) of the data. The testing set, on the other hand, will test the efficacy of the trained model on data it has never seen before. This divide allows the model to be neither overfit nor underfit and be able to perform accurately on new real-world data. At 80% we have enough data for training and at 20% we have enough data for a proper performance evaluation.

D) Normalize the Data:

Normalization is an important preprocessing step that puts all features on a similar scale, so that each feature can contribute fairly to the model's performance. This is typically done using the Standard Scaler, which normalizes the

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

data (subtracts the mean and scales to unit variance). That is, the average value of each feature will be 0, and its standard deviation will be 1. This can be very helpful for algorithms that need to work with feature scale since they won't be skewed by having larger variables. The Standard Scaler will make the dataset more consistent and make the model more stable and converge faster when training. This procedure finally improves the accuracy, efficiency, and generalization ability of the model on different datasets.

E) Algorithms:

DT is a supervised learning method that works by using a tree structure to make decisions based on the values of the features. It recursively divides the data using a splitting criterion that maximizes information gain or reduces impurity. DT may uncover the connections and relationships between the measurements and the failures for cloud task failure forecasting and offer comprehensible decision paths. Its aim is to develop a transparent rule-based model that explains the failure mechanisms, allowing for fast diagnostic insights and interpretability of the model, which is especially advantageous for operational monitoring and increase of the dependability in cloud environments [21].

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

KNN is a non-parametric, instance-based method that classifies a sample based on the majority class among its nearest neighbors. It measures the similarity by a function (e.g. Euclidean distance, Manhattan distance). This makes it useful for datasets with complex patterns and non-linear relationships. The advantage of KNN in cloud job failure prediction is that it is able to detect similar jobs to failed jobs in advance, enabling the identification of anomalies and proactive management of resources. Since KNN does not require a predefined data distribution, it can be useful for different cloud workload patterns [23].

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

ANN is a computational model inspired by biological neural networks and consists of interconnected layers of artificial neurons. It changes its weights in order to learn complex non-linear relationships in the process of training, through the back-propagation method. ANN is particularly suitable for high-dimensional data and complex patterns. In cloud job failure prediction, ANN can learn complex relationships between job features and failure risks to improve prediction performance. It aims at improving the resilience and adaptivity of failure detection in a wide range of cloud settings by applying deep feature learning [24].

The following equation is used to make the output prediction from the neural network.

Extreme Gradient Boosting (XGBoost) is an efficient implementation of gradient boosting, which constructs decision trees sequentially to correct the mistakes of previous DT. It enhances the performance with regularization, parallelization and sparsity awareness. XGBoost works very well with huge and structured data. For cloud job failure prediction, it enhances the reliability of models by capturing the subtle patterns of expected job failures. It is mainly oriented towards the development of classification accuracy through gradient based optimization with enhanced scalability and robustness in predictive modeling for large scale cloud systems [25].

The final ensemble prediction model of XGBoost is shown as follows.

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

AdaBoost is an ensemble learning method that combines multiple weak classifiers to form a stronger classifier. During training, greater attention is given to samples that were previously misclassified, thereby reducing the total error. It can improve classification performance, particularly when difficult-to-classify samples are present. In cloud task failure prediction, AdaBoost is proven to be useful in finding failure scenarios that are hard to detect by adaptively adjusting classifiers' weights. It aims to enhance the reliability and versatility of the model in order to correctly identify the failures even when the workloads are distributed differently and the computing environment is heterogeneous [27].

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

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

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

The Voting Classifier combines predictions from multiple base classifiers using a majority voting method (hard voting) or by averaging the probabilities (soft voting). It combines the different strengths of the individual algorithms to improve prediction stability. The voting method used in this paper is a combination of three algorithms: DT, KNN and ANN algorithms which have been used to enhance accuracy and reduce bias in cloud task failure prediction. The aim is to make "balanced" predictions under different conditions, allowing for consistent failure detection by taking advantage of the diversity of the classifiers that are involved [28].

In the case of classifiers, the equation below represents the majority vote.

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

The Stacking Classifier is an ensemble technique that combines multiple base classifiers with a meta-classifier to produce the final prediction. The outputs of the underlying models are aggregated by the meta-learner thus modelling the dependency on higher orders of the predictions. The base learners in this study are RF, KNN and MLP and as a meta-classifier Logistic Regression is used. Stacking can be helpful to increase generalization and predictive power. In this study, stacking combines the predictions of multiple algorithms to improve the accuracy and stability of cloud job failure prediction [30].The Meta-Model Prediction equation is:

$$\hat{y} = g(Y_{base}) = g\left(f_1(x), f_2(x), \ldots, f_m(x)\right) \qquad (7)$$

F) Integration of XAI & Flask Framework:

To enhance the interpretability and transparency of the ensemble prediction models, techniques were integrated from XAI like LIME and SHAP were adopted. They help users understand how individual input features influence the model's output., and learn the decision making processes of classic black-box models. XAI can support cloud failure management by improving the interpretability and transparency of model predictions through the visualization of feature importance and contribution values.

An interactive and lightweight web interface was developed using the Flask framework using the Flask framework to deploy the model in real-time. It enables entry of data, running predictions, and visibility of results in a web browser. By combining Flask with backend ML models, a scalable, efficient and user-friendly platform can be created to track and analyse the predictions of job failures in the cloud, on the fly.

IV. Experimental Results

Accuracy: Accuracy measures the proportion of correctly classified instances among all evaluated instances. It provides an overall measure of how accurately the model predicts the class of each cloud job. Accuracy is calculated as the proportion of correctly classified instances among all evaluated instances. This can be mathematically represented as:

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

Precision: Precision is the proportion of positive samples that are correctly identified from the samples that are classified as positive. Therefore, the precision formula is given as:

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

Recall: In ML, recall is a metric used to measure the ability of a model to identify all of the instances belonging to a specific class. It is the ratio of correctly predicted positive observations to the actual positive observations, and it is a measure of the completeness of the model with respect to capturing the instances of a particular class.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad (10)$$

F1-Score: F1-score is the harmonic mean of precision and recall. It provides a balanced measure of a classifier's ability to identify positive instances while limiting false-positive and false-negative errors. It's a combination of the precision and recall scores of a model. The F1 score provides a balanced measure based on precision and recall.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

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

"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

The results of accuracy, precision, recall and F1-score for cloud job failure predictions are displayed in Table 1, which shows that the stacking classifier is outperforming other classifiers in the overall performance.

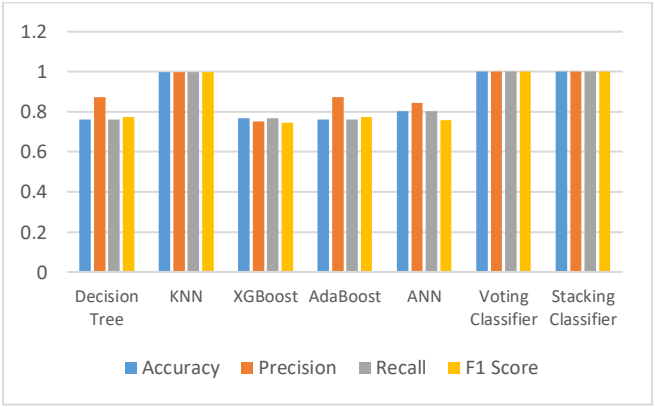


Figure.3 Performance Comparison of the Evaluated Models

Figure 3 compares the accuracy, precision, recall, and F1-score of the evaluated models. The results show that the Stacking Classifier achieved the highest overall performance among the evaluated approaches.



Figure.4 Enter input data

Figure 4. User-provided cloud job parameters submitted to the model for cloud job failure prediction.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

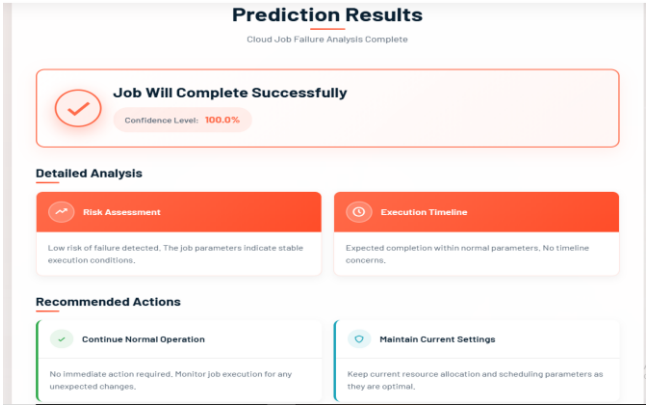


Figure.5 Prediction Output Indicating Successful Job Completion

Figure 5 shows the prediction output indicating successful job completion, along with the associated risk assessment and estimated execution information.



Figure.6 User-Provided Parameters for Failure Prediction

The right job parameters input to the system is shown in Figure 6 and is used for the system to process the input parameters and give the correct forecasts of cloud job failures.

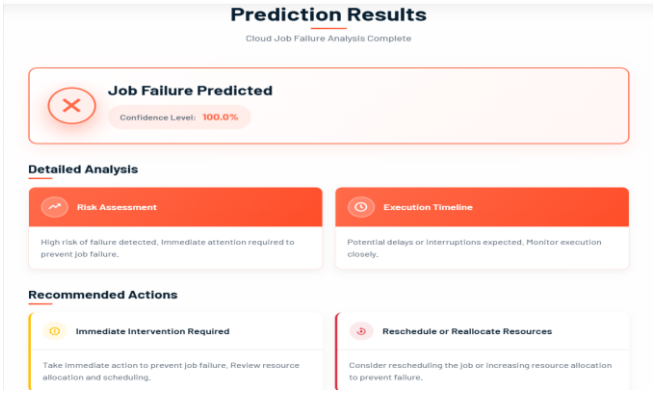


Figure.7 Prediction Output Indicating a Predicted Job Failure

Figure 7 shows the prediction output indicating "PREDICTION OUTPUT INDICATING A PREDICTED JOB FAILURE PREDICTED," along with the associated risk analysis and execution information.

V. CONCLUSION

The results show that a multilayer ensemble learning approach can be used to predict job failures in large-scale cloud systems using the Google Cluster 2019 dataset. Multiple classifiers are combined through voting and stacking to improve prediction of performance and model robustness. The ensemble model with voting strategy of DT, KNN, and ANN achieved an excellent accuracy of 97.83% in the job failure classification. The stacked model, which uses RF, KNN, and MLP with Logistic Regression as the meta-estimator, achieved a perfect prediction accuracy of 100%. These results highlight the strength and adaptability of ensemble methods for predicting cloud performance in challenging and

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

complex situations. Moreover, the main contributors for the prediction results were identified using XAI techniques for better interpretability, including LIME and SHAP. The framework provides an interpretable approach to cloud job failure prediction and has the potential to support fault tolerance and operational efficiency in cloud data centers.

Further research is needed to enhance the multilayer ensemble architecture to handle real-time streams and heterogeneous cloud systems. For temporal and contextual aspects, RNN and transformer models can be used to enhance failure prediction. The framework could be deployed on a container orchestration platform such as Kubernetes to support adaptation to dynamic workloads. Federated learning could be explored to enable model training across multiple data centers while reducing the need to share raw data. Future XAI implementations could provide more detailed visual explanations and improve the interpretability of model predictions. Future work on failure severity prediction and self-healing mechanisms could support more autonomous and reliable cloud management systems.

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.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 10, October 2026

DOI: 10.17148/IARJSET.2026.131044

[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.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License