HIGH-ACCURACY CLOUD JOB STATUS CLASSIFICATION WITH CATBOOST AND ENSEMBLE VOTING TECHNIQUES

SK. HimamBasha [1], Y SaiPavanKumar [2]
Assistant Professor [1], PG Scholor [2]
Department of Master of Computer Applications
QIS College of Engineering & Technology (Autonomous), Ongole
Prakasam Dist., AP

Abstract: For system reliability, resource efficiency, and continuous service delivery, modern cloud computing systems need accurate task failure prediction. In order to increase accuracy and robustness, this study incorporates CatBoost, a powerful gradient boosting algorithm, into a multilayer voting-based prediction framework. Compared to standard models, CatBoost is better at handling categorical features and complex data distributions in cloud workloads. Its symmetric tree structure and integrated overfitting control facilitate generalization in a variety of dynamic scenarios.

Outperforming machine learning and ensemble approaches, the improved system uses the Google Cluster 2019 trace dataset to train and assess the model. Users may submit task data and instantly obtain failure predictions and classification results using a Flask-based web interface, which further enhances the model. This integration makes theoretical models applicable to real-world situations.

The results of experiments show that the CatBoost-based extension works better than competing framework classifiers and produces reliable results even on unknown data. The proposed enhancement improves the accuracy, scalability, and usefulness of the cloud task failure prediction system, making it suitable for cloud infrastructure management and decision-making.

Index terms - *cloud job failure prediction, multilayer voting classifier, ensemble learning, CatBoost, Random Forest, XGBoost, ANN.*

1. INTRODUCTION

computational resources by offering scalable, flexible, and on-demand services via the Internet. As organizations shift more and more workloads, apps, and critical processes to cloud platforms, maintaining continuous service availability has become critical to contemporary IT architecture. Cloud providers must offer consistent performance, high availability, and seamless user job execution because any disruption directly affects service quality and user experience. However, because of the rapid expansion and growing complexity of cloud systems, resource management, workload optimization, and system reliability now confront significant challenges. Job failures in the cloud can be caused by a variety of factors, including scheduling conflicts, software bugs, hardware issues, and resource limitations. These issues not only reduce service quality but also waste a lot of processing and storage power. Events in the real world, such as the extensive outages experienced by major service providers, demonstrate how important early failure detection is to preventing service disruptions and maintaining operational continuity. As the number and variety of cloud workloads continue to increase, it has become essential to comprehend task execution characteristics and identify potential problems before they occur in order to provide dependable cloud services. Our study focuses on increasing the effectiveness and precision of cloud task failure prediction to deliver more reliable and efficient cloud operations.

2. LITERATURE SURVEY

1. Forecasting Cloud Application Workloads With CloudInsight for Predictive Resource Management:

To get around the drawbacks of reactive cloud autoscaling, predictive cloud resource management has become popular. Workload predictors, which forecast short- and long-term variations in cloud application workloads, are a major component of predictive resource management. These predictors are often pre-optimized for particular patterns of workload. These predictors, however, are still insufficient to manage real-world cloud workloads, whose patterns may be irregular, dynamic, and unknown beforehand. Because of this, these factors frequently result in either excessive or insufficient cloud resource allocation. Using the combined strength of several workload predictors, we have developed CloudInsight, a unique cloud workload prediction framework, to solve this issue. To produce precise forecasts for actual workloads, CloudInsight builds an ensemble model with several predictors. Using multi-class regression, the weights of the predictors in CloudInsight are calculated at runtime based on their accuracy for the present workload. Periodically, the ensemble model is adjusted to accommodate abrupt shifts in the workload. We used a variety of actual workload traces to assess CloudInsight. According to the findings, CloudInsight outperforms the most advanced predictors by 13–27%. Additionally, trace-based simulations using a cloud resource management reveal that CloudInsight has 15–20% fewer under-/over-provisioning periods, which leads to minimal SLA breaches and good cost-efficiency.

2. A Fault Tolerant Elastic Resource Management Framework Toward High Availability of Cloud Services:

Every digital service now requires cloud computing, which has led to an exponential rise in its use. However, a massive increase in the demand for cloud resources prevents service availability, which leads to load imbalance, performance degradation, outages, and excessive power consumption. The current methods mostly try to solve the issue by employing several clouds and running numerous virtual machine (VM) copies, which results in significant operating costs. In order to handle the aforementioned issue from a different angle, this study suggests a Fault Tolerant Elastic Resource

Management (FT-ERM) architecture that induces high-availability in servers and virtual machines. In particular, (1) an online failure predictor is created to predict failure-prone virtual machines (VMs) based on anticipated resource contention; (2) the operational status of the server is monitored using a power analyzer, resource estimator, and thermal analyzer to detect any failure caused by overloading and overheating of servers proactively; and (3) failure-prone VMs are assigned to a proposed fault-tolerance unit made up of a decision matrix and safe box to trigger VM migration and handle any outages in advance while maintaining the desired level of availability for cloud users. Two real-world datasets are used in experiments to assess and compare the suggested framework with the state-of-the-art. In comparison to the non-FT-ERM strategy, FT-ERM reduced VM migration and power consumption by 88.6% and 62.4%, respectively, and increased service availability by 34.47%.

3. A Deep Dive into the Google Cluster Workload Traces: Analyzing the Application Failure Characteristics and User Behaviors

Because of their high availability, quick flexibility, scalability, and affordability, large-scale cloud data centers have grown in popularity. However, due to improper resource use and early failure detection, data centers nowadays still have significant failure rates. Understanding the workload and failure characteristics is essential for optimizing resource efficiency and lowering failure rates in large-scale cloud data centers. The 2019 Google Cluster Trace Dataset, which comprises 2.4TiB of workload traces from eight distinct clusters worldwide, is thoroughly analyzed in this study. In Google's production cloud, we examine the features of failed and terminated jobs and try to link them with important variables including resource utilization, job priority, scheduling class, job length, and the quantity of task resubmissions. Our study identifies a number of significant traits of unsuccessful tasks that may be utilized to create an early failure prediction system. Additionally, we offer a unique use analysis to find heterogeneity in user-submitted tasks and jobs. On a single cluster, we are able to pinpoint individual users who are in charge of almost half of all collection events. We argue that these features might be helpful in creating an early task failure prediction system that could be used to dynamically reschedule the job scheduler, enhancing resource utilization and lowering failure rates in large-scale cloud data centers.

4. Proactive Fault-Tolerance Technique to Enhance Reliability of Cloud Service in Cloud Federation Environment

When their resource demands are low, cloud service providers (CSPs) may now give their idle virtual machines to other CSPs thanks to a new computing model called cloud federation. When their needs for computer resources are enormous, federation also enables CSPs to outsource their resource requirements to other CSPs. when a result, when CSPs are able to pool their resources, the availability and dependability of services provided by service providers rise in cloud federation environments. Furthermore, the computational environment of member CSPs in the federation must be fault tolerant in order to preserve the availability and dependability of cloud services provided through federation. Fault-tolerant systems are therefore required in cloud federation environments to ensure cloud service availability and dependability. In this paper, we present a proactive fault tolerance system that uses CPU temperature to prevent failures inside the federation. Redistributing resources (virtual machines) from faulty CSPs to non-faulty CSPs within the federation while increasing profit and decreasing migration costs is how the fault tolerance mechanism inside the federation is depicted. We have also suggested an algorithm known as Preference Based Fault Management (PBFM) to handle the federation in the case of faults in order to overcome this problem. We conduct comprehensive tests to assess the efficacy of our suggested method and contrast it with two other mechanisms: PAFM (Profit Assured Fault Management) and MCAFM (Migration Cost Assured Fault Management). The findings demonstrate that, in the face of defective CSPs, our suggested mechanism PBFM provides an optimal solution to the general problem of profit and migration cost trade-off.

5. From binary to quaternary copper chalcogenide compounds in solar cells technology: Recent progress and perspectives

For a variety of electrical applications, including solar cells, copper chalcogenides (CuCh, where Ch = O, S, Se, and Te) are potential semiconductor materials. CuInGaSe2 (CIGS), one of these substances, is a great option for obtaining high power conversion efficiency (PCE) in solar cell applications. CuCh-based thin-film solar cells provide a reliable and affordable substitute for conventional Si-based solar cells with similar efficiency. Some CuCh compounds have been used in perovskite and dye-sensitized solar cells (DSSC) because of their exceptional semiconducting and optoelectronic qualities. Nevertheless, there are still a number of CuCh compounds that have great promise for usage in solar cells but have not yet been thoroughly investigated. An overview of the present status of CuCh-based solar cells is given in this review article, which covers the basic ideas, manufacturing processes, and technologies used in their creation. The study encompasses less developed compounds that have not yet been thoroughly investigated for solar cell technology, in addition to well-known CuCh photovoltaic materials. Along with addressing important issues related to the commercialization and marketability of this material, its potential use in future consumer products, and considerations for solar cell waste management, the article also outlines workable strategies to increase the viability of CuCh-based solar cells in society.

3. METHODOLOGY

i) Proposed Work:

The suggested work integrates CatBoost, an advanced gradient boosting algorithm, into the multilayer voting-based cloud task failure prediction framework to increase forecast accuracy and robustness. The system is designed as a hybrid, multi-stage architecture that combines ensemble learning, deep learning, and boosting techniques to effectively manage complex cloud workload patterns. In order to manage missing values, standardize features, and prepare categorical characteristics for efficient learning, the Google Cluster 2019 dataset is first preprocessed. The first step is utilizing GridSearch to train and optimize a number of fundamental classifiers, including as Decision Tree, K-Nearest Neighbors (KNN), AdaBoost, XGBoost, and Artificial Neural Networks (ANN), in order to produce probability-based predictions. These results are combined using a weighted soft voting approach to produce a reliable classification of task status (success or failure).

To further strengthen the system, CatBoost, an improved high-performance model trained on the same dataset, is introduced. Because CatBoost naturally handles categorical data, reduces overfitting, and captures complex feature interactions, it outperforms traditional models in terms of prediction. In the second stage, the identified failed jobs are categorized into several failure types, such as Lost, Kill, Evict, Finish, and Fail, using a Random Forest classifier to enable a deeper understanding of system behavior. Additionally, a Flask-based web application is developed to provide real-time prediction capabilities. Users may immediately receive prediction results by submitting job-related data through an interactive portal. All things considered, the proposed work provides an extremely accurate, scalable, and user-friendly method for forecasting cloud task failures, improving system reliability, and enabling proactive resource management in cloud systems.

ii) System Architecture:

To accurately anticipate job failure and type, the multilayer prediction framework successively processes cloud job data. The Job Dataset (Google Cluster 2019), which includes task behavior, scheduling, and resource usage, is the first input. To handle missing values, normalize features, and get categorical characteristics ready for model training, data is preprocessed. The processed dataset is sent to the first prediction layer, which is the system's primary decision-maker.

In the first layer, voting classifiers employ Artificial Neural Networks, AdaBoost, K-Nearest Neighbors (KNN), and Decision Trees. Each model predicts whether a job will succeed or fail on its own. To produce a more reliable and consistent outcome, a weighted soft voting strategy integrates all classifier strengths. CatBoost, a strong gradient boosting model that efficiently manages category data and complex feature interactions to provide accurate predictions, is added to the voting framework.

Unsuccessful tasks are classified as Lost, Kill, Evict, Finish, or Fail by a Random Forest classifier in the second layer, which receives the output from the first layer. Errors are identified and their causes are explained using this tiered approach. Lastly, the system outputs the Job Failure Type for fault management, resource optimization, and proactive decision-making.

Additionally, the design has a Flask-based web interface for real-time user interaction, which makes it deployment-ready by enabling users to provide input data and get prediction results.

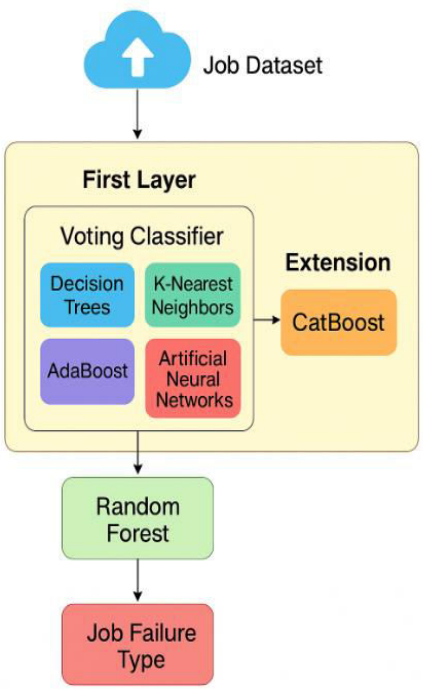

Fig.1. Proposed Architecture

iii) MODULES:

1. Importing the Packages

All necessary Python libraries for data processing, visualization, machine learning, deep learning, CatBoost, and Flask deployment are initialized in this module. It guarantees that preprocessing, model training, assessment, and web integration operations are carried out smoothly.

2. Exploring the Dataset

In order to comprehend feature kinds, task behavior, and failure patterns, this module loads and examines the Google Cluster dataset. It aids in finding discrepancies and directs model selection and preprocessing.

3. Visualization

This lesson examines the distribution of employment success and failure using graphical methods such as bar charts. It assists in identifying unbalance and comprehending how attributes affect task outcomes.

4. Pre-Processing

By addressing missing values, eliminating inconsistencies, and normalizing features, this module cleans the dataset. It guarantees higher model performance and enhances data quality.

5. Label Encoding

This module uses encoding techniques to translate category information into numerical values. It makes it possible for machine learning models to effectively handle non-numeric inputs.

6. Data Splitting (Train & Test)

The dataset is divided into training and testing sets (80–20) using this module. It avoids overfitting and guarantees appropriate model validation.

7. Model Training

Several models, such as Decision Tree, KNN, XGBoost, AdaBoost, ANN, Random Forest, and CatBoost, are trained by this module. While CatBoost improves accuracy with higher generalization, a voting classifier combines predictions.

8. Evaluation

Accuracy, precision, recall, and F1-score are used in this module's model evaluation. It verifies prediction reliability and compares performance.

9. Flask Server

This module creates a web-based interface by deploying the learned model using Flask. Real-time prediction and user engagement are made possible by it.

10. User Login

This module uses authentication to provide safe access. The prediction system is only accessible by authorized users.

11. Job Failure & Type Prediction

This module takes user input, uses voting/CatBoost models to forecast work status, and uses Random Forest to categorize failure types. For decision-making, the results are presented in an understandable manner.

12. Logout

After use, this module safely terminates the user session and stops unwanted access.

iv) ALGORITHMS:

1. The Decision Tree

A supervised learning technique called Decision Tree divides data into hierarchical branches according to feature requirements in order to make a decision. A condition is represented by each internal node, whereas final results are represented by leaf nodes. It functions well with both numerical and categorical data and is quite interpretable. This system classifies jobs as successful or unsuccessful and assists in identifying rule-based trends in cloud job execution. After training, it makes quick predictions and is computationally efficient.

2. KNN, or K-Nearest Neighbors

KNN is a non-parametric method that uses distance metrics such as Euclidean distance to classify data according to similarity. The majority class among the closest neighbors is used to determine the class label. For datasets with distinct patterns, this approach is straightforward and efficient. To ascertain anticipated outcomes, cloud job prediction contrasts fresh job inputs with past job data. Although it adjusts well to pattern-based categorization, it could need more processing power for big datasets.

3. XGBoost

XGBoost is a sophisticated gradient boosting method that generates models one after the other, fixing the mistakes of the preceding model. It incorporates regularization strategies to enhance generalization and avoid overfitting. It is scalable for big datasets and very effective. XGBoost enhances prediction accuracy in this system by capturing intricate correlations between job characteristics. Additionally, it offers feature significance insights and efficiently manages missing data.

4. AdaBoost

AdaBoost is an ensemble learning technique that creates a strong classifier by combining several weak learners. In each cycle, it gives misclassified samples more weight, thereby increasing accuracy. Prediction mistakes are decreased by this adaptive learning technique. AdaBoost improves classification in cloud job prediction by concentrating on challenging scenarios. By repeatedly fixing errors from earlier models, it improves dependability.

5. ANNs, or artificial neural networks

An artificial neural network (ANN) is a deep learning model with input, hidden, and output layers that is modeled after the human brain. It learns intricate patterns by processing input using activation functions and weighted connections. Non-linear connections in high-dimensional data can be captured using ANN. It aids in the analysis of complex relationships between job characteristics that affect failure in this system. Through ongoing learning and improvement, it offers great predicted accuracy.

6. Proposed Layer-1 Ensemble Voting Model

The ensemble voting model uses a weighted soft voting method to combine predictions from several classifiers. Probability ratings are contributed by each model, and the aggregate results are used to make the ultimate conclusion. This method increases stability while lowering bias and volatility. It functions as the initial layer in the suggested system to categorize jobs as successful or unsuccessful. By using the complimentary qualities of many algorithms, it improves resilience.

7. Layer-2 Classifier: Random Forest

Using random selections of data and attributes, the Random Forest ensemble approach creates many decision trees. A majority vote among all trees determines the ultimate result. It enhances prediction stability and lessens overfitting. It is employed in the second layer of this framework to categorize particular instances of job failure. It also aids in determining key characteristics that impact failure behavior.

8. Extension Model CatBoost

A gradient boosting method called CatBoost was created especially to effectively handle category data. To avoid overfitting and enhance generalization, it makes use of symmetric trees and ordered boosting. It performs well on complicated datasets with little preparation. CatBoost functions as an enhanced model that provides better prediction accuracy in this system. It is very dependable for failure prediction since it successfully catches intricate patterns in cloud workloads.

4. EXPERIMENTAL RESULTS

The experimental findings show how several machine learning and deep learning algorithms for cloud job failure prediction compare in terms of performance. Key measures including Accuracy, Precision, Recall, and F1-Score are used in the evaluation. The majority of the models, including Decision Tree, ANN, KNN, XGBoost, and the suggested ensemble model, achieve near-perfect performance with values close to 99–100%, according to the results, suggesting high learning potential and efficient feature use.

The CatBoost (Extension Model) outperforms all other algorithms and demonstrates its supremacy in processing complicated and categorical cloud workload data by achieving 100% accuracy, precision, recall, and F1-score. Additionally, the suggested ensemble voting model shows the benefit of merging several classifiers by producing extremely stable and consistent results. AdaBoost, on the other hand, performs quite poorly, underscoring its shortcomings in managing intricate data patterns. Overall, the findings demonstrate that the suggested multilayer architecture with CatBoost extension greatly enhances cloud job failure prediction systems' prediction accuracy, resilience, and dependability.

Accuracy: The ability of a test to differentiate between healthy and sick instances is a measure of its accuracy. Find the proportion of analysed cases with true positives and true negatives to get a sense of the test's accuracy. Based on the calculations:

Accuracy = TP + TN /(TP + TN + FP + FN)

$$Accuracy = \frac{(TN + TP)}{T}$$

Precision: The accuracy rate of a classification or number of positive cases is known as precision. Accuracy is determined by applying the following formula:

Precision = True positives/ (True positives + False positives) = TP/(TP + FP)

$$\Pr e\,cision = \frac{TP}{(TP + FP)}$$

Recall: The recall of a model is a measure of its capacity to identify all occurrences of a relevant machine learning class. A model's ability to detect class instances is shown by the ratio of correctly predicted positive observations to the total number of positives.

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

F1-Score: A high F1 score indicates that a machine learning model is accurate. Improving model accuracy by integrating recall and precision. How often a model gets a dataset prediction right is measured by the accuracy statistic..

$$F1 = 2 \cdot \frac{(Recall \cdot \Pr e\,cision)}{(Recall + \Pr e\,cision)}$$

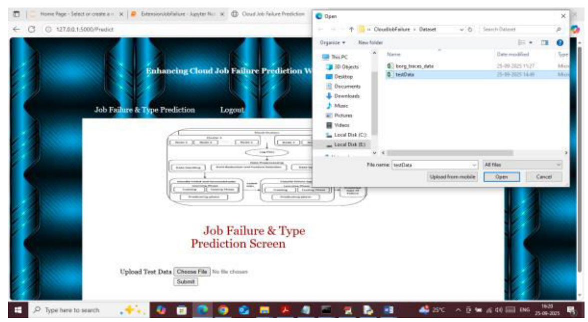

Figure 2: Test Data Upload Module

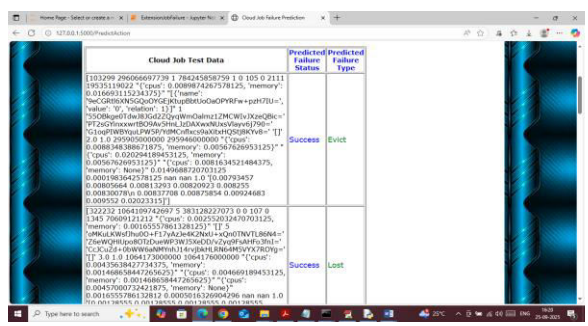

Figure 3: Prediction Results

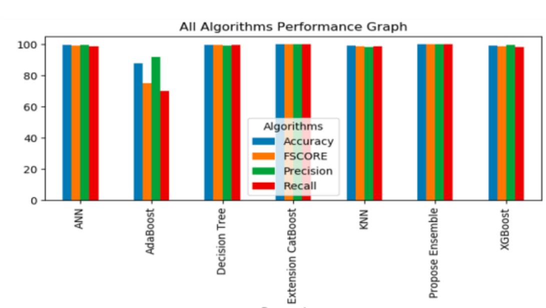

Figure 4: Accuracy Graph

5. CONCLUSION

In order to increase accuracy and dependability in challenging cloud settings, this research proposes an effective multilayer voting-based cloud task failure prediction system. Numerous machine learning and deep learning models are used in the first layer to capture different patterns of task execution behavior. While second-layer Random Forest helps comprehend system performance by classifying failure kinds, weighted soft voting enhances forecast stability.

The model is strengthened with the CatBoost extension, which enhances accuracy, generalization, and categorical data processing. The expanded model predicts almost completely better than conventional methods, according to experimental data. The system is easy to use for real-time applications thanks to a Flask-based web interface. All things considered, the suggested system for resource management and cloud task failure prediction is reliable, expandable, and deployable.

6. FUTURE SCOPE

Real-time streaming data processing using Apache Kafka or Spark can enhance continuous monitoring and quick failure prediction in live cloud systems. It would stop work failures and increase responsiveness. To enhance the prediction of sequential workload patterns, deep learning architectures such as Transformer models or LSTM may identify temporal correlations in task execution data.

For the scalability and adaptability of cloud platforms, future work may implement the system in distributed and multi-cloud environments. Resource optimization and forecast-based auto-scaling improve dynamic workload management. The usability of online interfaces will be enhanced by dashboard visualization and alarm systems, enabling managers to make decisions about cloud reliability more quickly and intelligently.

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. Int. J. Eng. Res. Technol.(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] I. K. Kim, W. Wang, Y. Qi, and M. Humphrey, ''Forecasting cloud appli cation workloads with CloudInsight for predictive resource management,'' IEEE Trans. Cloud Comput., vol. 10, no. 3, pp. 1848–1863, Jul. 2022, doi: 10.1109/TCC.2020.2998017.

[7] D. Saxena, I. Gupta, A. K. Singh, and C.-N. Lee, ''A fault tolerant elastic resource management framework toward high availability of cloud services,'' IEEE Trans. Netw. Service Manage., vol. 19, no. 3, pp. 3048–3061, Sep. 2022.

[8] F. H. Bappy, T. Islam, T. S. Zaman, R. Hasan, and C. Caicedo, ''A deepdive into the Google cluster workload traces: Analyzing the application failure characteristics and user behaviors,'' in Proc. 10th Int. Conf. Future Internet Things Cloud (FiCloud), Aug. 2023, pp. 103–108.

[9] B. K. Ray, A. Saha, S. Khatua, and S. Roy, ''Proactive fault-tolerance technique to enhance reliability of cloud service in cloud federation envi ronment,'' IEEE Trans. Cloud Comput., vol. 10, no. 2, pp. 1–12,Apr. 2022, doi: 10.1109/TCC.2020.2968522.

[10] X. Chen, L. Yang, Z. Chen, G. Min, X. Zheng, and C. Rong, ''Resource allocation with workload-time windows for cloud-based software services: A deep reinforcement learning approach,'' IEEE Trans. Cloud Comput., vol. 11, no. 2, pp. 1–15, Apr. 2023, doi: 10.1109/TCC.2022.3169157.

[11] A. Priyadarshini, S. K. Pradhan, S. R. Laha, and D. S. K. Nayak, ''Enhancing fault tolerance and load balancing in cloud computing for improved e-healthcare systems performance,'' in Proc. 2nd Int. Conf. Ambient Intell. Health Care (ICAIHC), Nov. 2023, pp. 1–6, doi: 10.1109/icaihc59020.2023.10431436.

[12] S. Chouliaras and S. Sotiriadis, ''An adaptive auto-scaling framework for cloud resource provisioning,'' Future Gener. Comput. Syst., vol. 148, pp. 173–183, Nov. 2023, doi: 10.1016/j.future.2023.05.017.

[13] K. Vani and S. Sujatha, ''A machine learning framework for job failure prediction in cloud using hyper parameter tuned MLP,'' in Proc. 2nd Int. Conf. Adv. Technol. Intell. Control, Environ., Comput. Commun. Eng. (ICATIECE), Dec. 2022, pp. 1–6, doi: 10.1109/ICATIECE56365.2022.10047809.

[14] M. M. A. Moni, M. Niloy, M. F.-U.-A. Juboraj, and A. Banik, ''AI based cloud failure detection and prevention algorithm,'' in Proc. Innov. Power Adv. Comput. Technol. (i-PACT), Kuala Lumpur, Malaysia, Dec. 2023, pp. 1–5, doi: 10.1109/i-pact58649.2023.10434295.

[15] K. Vani and S. Sujatha, ''Enhancing fault tolerance system in cloud environment through failure prediction using ensemble learning,'' in Proc. 2nd Int. Conf. Intell. Data Commun. Technol. Internet Things (IDCIoT), Jan. 2024, pp. 851–858, doi: 10.1109/idciot59759.2024. 10467215.

[16] V. Salanke and V. Kowshik, ''Task failure prediction and migration in cloud environment,'' in Proc. 1st Int. Conf. Commun. Comput. Sci. (InCCCS), May 2024, pp. 1–6, doi: 10.1109/InCCCS60947.2024.10593580.

[17] S. Jomah and S. Aji, ''Google cloud trace: Characterization of terminated jobs,'' in Proc. 2nd Int. Conf. Advancement Comput. Comput. Technol. (InCACCT), May 2024, pp.517–522, doi: 10.1109/incacct61598.2024.10551146. 140612

[18] A.-I. Tuns and A. Spataru, ''Cloud service failure prediction on Google's Borg cluster traces using traditional machine learning,'' in Proc. 25th Int. Symp. Symbolic Numeric Algorithms Sci. Comput. (SYNASC), Sep. 2023, pp. 162–169, doi: 10.1109/synasc61333.2023.00029.

[19] A. Ulah, Z. Alomari, S. Alkhushayni, and D. Al-zaleq, ''Improvement in task allocation for VM and reduction of Makespan in IaaS model for cloud computing,'' Cluster Comput., vol. 27, pp. 11407–11426, Jul. 2024, doi: 10.1007/s10586-024-04539-8.

[20] J. Gao, H. Wang, and H. Shen, ''Task failure prediction in cloud data centers using deep learning,'' IEEE Trans. Services Comput., vol. 15, no. 3, pp. 1411–1422, May 11, 2022, doi: 10.1109/TSC.2020.2993728.

[21] R. R. Swain, D. S. K. Nayak, and T. Swarnkar, ''A comparative analysis of machine learning models for colon cancer classification,'' in Proc. Int. Conf. Adv. Power, Signal, Inf. Technol. (APSIT), Jun. 2023, pp. 1–5, doi: 10.1109/apsit58554.2023.10201691.

[22] D. Y. Omkari and K. Shaik, ''An integrated two-layered voting (TLV) framework for coronary artery disease prediction using machine learn ing classifiers,'' IEEE Access, vol. 12, pp. 56275–56290, 2024, doi: 10.1109/ACCESS.2024.3389707.

[23] A. Arabiat, M. Hassan, and O. Almomani, ''WEKA-based machine learn ing for traffic congestion prediction in amman city,'' IAES Int. J. Artif. Intell. (IJ-AI), vol. 13, no. 4, p. 4422, Dec. 2024.

[24] Zhao, J., Ding, Y., Zhai, Y., Jiang, Y., Zhai, Y., & Hu, M. (2021). Explore unlabeled big data learning to online failure prediction in safety-aware cloud environment. Journal of Parallel and Distributed Computing, 153, 53-63.

[25] Gollapalli, M., AlMetrik, M. A., AlNajrani, B. S., AlOmari, A. A., AlDawoud, S. H., AlMunsour, Y. Z., ... & Aloup, K. M. (2022). Task Failure Prediction Using Machine Learning Techniques in the Google Cluster Trace Cloud Computing Environment. Mathematical Modelling of Engineering Problems, 9(2).

[26] Jassas, M. S., & Mahmoud, Q. H. (2021, April). A failure prediction model for large scale cloud applications using deep learning. In 2021 IEEE International Systems Conference (SysCon) (pp. 1-8). IEEE.

[27] Vani, K., & Sujatha, S. (2024, January). Enhancing Fault Tolerance System in Cloud Environment Through Failure Prediction Using Ensemble Learning. In 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT) (pp. 851-858). IEEE.

[28] Malleswaran, S. K. A., Marimuthu, K., Robert, P., & Kuppan, P. (2025). Optimized Bayesian tensorized neural network affording task failure prediction in cloud environment. Expert Systems with Applications, 285, 127538.

[29] Vani, K., & Sujatha, S. (2022, December). A machine learning framework for job failure prediction in cloud using hyper parameter tuned MLP. In 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE) (pp. 1-6). IEEE.

[30] Tuns, A. I., & Spătaru, A. (2023, September). Cloud Service Failure Prediction on Google's Borg Cluster Traces Using Traditional Machine Learning. In 2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC) (pp. 162-169). IEEE.

Author profiles

Mr. HIMAMBASHA Shaik is an Assistant Professor in the Department of Master of Computer Applications at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He earned his Master of Computer Applications (MCA) from Anna University, Chennai. With a strong research background, He has authored and co-authored research papers published in reputed peer-reviewed journals. His research interests include Machine Learning, Artificial Intelligence, Cloud Computing, and Programming Languages. He is committed to advancing research and fostering innovation while mentoring students to excel in both academic and professional pursuits.

Mr. Y. SAI PAVAN KUMAR is an MCA Student in the Department of Computer Application at QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He has Completed Degree in B.Com.(General) from D.C.R.M Degree College Inkollu, Bapatla district