IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 1 (2026)

ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 1, 2026

Peer Reviewed, Referred & Indexed Journal

Research

High-Accuracy Cloud Job Failure Prediction Using a Two-Layer Voting Architecture

Mrs. G. BALESWARI1, KASINIKOTA GNANA SAI ROHITH2, ATTI RAJASEKHAR3, CHERUKURI CHARAN TEJA4

#1 Associate Professor , Department of Information Technology , NRI Institute of Technology Agiripalli.

#2#3#4 B.Tech with Specialization of Information Technology in NRI Institute Of Technology,Agiripalli

Abstract: Modern cloud computing systems require precise job failure prediction for system dependability, resource efficiency, and ongoing service delivery. This study adds CatBoost, a strong gradient boosting algorithm, to a multilayer voting-based prediction framework to improve accuracy and resilience. CatBoost handles categorical features and complicated data distributions in cloud workloads better than traditional models, and its symmetric tree structure and built-in overfitting control promote generalization across varied and dynamic contexts.

The enhanced system trains and evaluates the model using the Google Cluster 2019 trace dataset, outperforming machine learning and ensemble methods. In addition to model improvement, a Flask-based web interface allows users to upload task data and instantaneously receive failure predictions and categorization results. This integration connects theoretical models to practice.

Experimental findings reveal that the CatBoost-based extension outperforms competing framework classifiers and delivers dependable results even on unseen data. The suggested addition enhances the cloud job failure prediction system's accuracy, scalability, and usability, making it appropriate for cloud infrastructure management and decision-making.

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

Received: 31-01-2026 Accepted: 05-03-2026 Published: 12-03-2026

1. INTRODUCTION

computing resources by using the Internet to provide scalable, adaptable, and on-demand services. Maintaining continuous service availability has become essential to modern IT infrastructure as businesses move more and more workloads, apps, and crucial functions to cloud platforms. Since each interruption has a direct impact on service quality and user experience, cloud providers are required to provide steady performance, high availability, and smooth user job execution. However, resource management, workload optimization, and system stability now face major issues due to the cloud environments' quick growth and increasing complexity. A number of things, such as schedule conflicts, software flaws, hardware malfunctions, and resource shortages, can cause job failures in the cloud. These malfunctions result in significant waste of processing and storage power in addition to lowering service quality. Real-world events, like the widespread outages that major service providers have seen, highlight how crucial early failure detection is to averting service interruptions and preserving operational continuity. Understanding task execution characteristics and spotting possible errors before they happen have become crucial for guaranteeing reliable cloud services as the volume and diversity of cloud workloads keep growing. In order to provide more robust and productive cloud operations, our work focuses on improving the efficacy and accuracy of cloud task failure prediction.

2. LITERATURE SURVEY

1. Analysis of Job Failure and Prediction Model for Cloud Computing Using Machine Learning:

Modern applications like eHealth, home automation, and smart cities necessitate a new strategy to increase the availability and dependability of cloud applications. The majority of cloud services, including software and hardware, have experienced malfunctions because of the vastness and variety of the cloud environment. In this work, we first use publicly available traces to examine and describe the behavior of completed and unsuccessful tasks. To identify unsuccessful jobs before they happen, we have created a failure prediction model. The suggested methodology seeks to improve cloud application efficiency and resource usage. We assess the suggested model using three publicly accessible traces: Trinity, Mustang, and the Google cluster. To determine which machine learning model was the most accurate, the traces were also run through a number of other models. Our findings show a strong relationship between tasks that failed and the resources that were sought. Our model's high accuracy, recall, and F1-score were also demonstrated by the evaluation results. The availability and dependability of cloud services may be increased by a number of strategies, including restricting task re-submission, creating scheduling algorithms, altering priority policies, and forecasting work failure.

2. Machine learning job failure analysis and prediction model for the cloud environment:

The future of smart appliances, electronic health, and ubiquitous computing depends on dependable and easily available cloud apps. Most cloud services, both logical and physical, have failed because of the cloud's size and variety. We evaluated and described the behaviors of successful and failed actions using traces that are already available. We came up with and put into practice a technique to predict which jobs will fail. The suggested approach improves the resource utilization efficiency of cloud applications. A thorough assessment of the suggested model was carried out using publicly accessible Google Cluster, Mustang, and Trinity traces. To choose the most dependable model, the traces were also put into a number of machine learning models. Our efficiency study demonstrates that the model's accuracy, F1-score, and recall are all high. The reliability and availability of cloud services may be increased by a number of variables, including predicting failure, scheduling algorithm design, priority criteria adjustment, and task resubmission restrictions.

3. Prediction of Cloud Server Job Failures using Machine Learning based KNN Classification and LSTM Modelling Methods

Instead of using a local server or a personal computer to store, manage, and analyze data, cloud computing uses a network of distant servers housed online. Since the cloud computing sector has expanded significantly in recent years, it is crucial to ensure that the service being provided does not diverge from the planned offering. So, to build a reliable cloud service platform, we need to understand and characterize failures. Understanding the causes of failure and forecasting potential future failures are the goals of this endeavor. We utilize the Google cluster workload trace to do this. According to our data, jobs that ultimately fail require a disproportionate amount of resources. In order to anticipate the qualities that are essential to failure, we employ Long Short-Term Memory Networks (LSTM). The KNN classification model, which was trained on the available data, is then used to forecast the termination status.

4. Cloud failure prediction based on traditional machine learning and deep learning

One of the most important problems is cloud failure, which may cost cloud service providers millions of dollars in addition to the productivity losses experienced by industrial users. The main strategy to deal with this problem is fault tolerance management, and one method to stop a failure from happening is failure prediction. Creating a highly accurate predictive model is one of the primary issues in failure prediction. A thorough assessment of models based on various machine learning methods is currently lacking, despite some work on failure prediction models having been offered. Consequently, we present a thorough comparison and model evaluation of prediction models for task and job failure in this work. Three deep learning algorithm variations and five conventional machine learning algorithms are used in the construction and training of these models. Google Cloud Traces is a benchmark dataset that we utilize to train and evaluate the models. We tested the scalability of the models, identified their key characteristics, and assessed their performance using a variety of measures. The following conclusions were drawn from our analysis. First, we discovered that Extreme Gradient Boosting yields the best model for task failure prediction, with the disk space and CPU requests being the key factors affecting the forecast. Second, we discovered that Random Forest and Decision Tree generate the best models for task failure prediction, with the task priority being the most crucial component for both models. The Logistic Regression model is the most scalable among the others, according to our scalability research.

5. Workload Failure Prediction for Data Centers

The efficiency of HPC data centers is greatly impacted by failed workloads that used a large amount of time and space for calculation, which in turn limits the amount of scientific work that can be accomplished. Although computing power has grown dramatically over time, workload failure detection and prediction have trailed far behind and will become more crucial as system complexity and scale continue to rise. In order to forecast workload failures, we train machine learning models on a large number of data sets and examine workload traces gathered from a production cluster. Our prediction models include a runtime model that forecasts failures at runtime and a queue-time model that calculates the likelihood of workload problems prior to execution. According to evaluation findings, workload failures may be predicted by the queue-time model and the runtime model with a maximum precision score of 90.61% and 97.75%, respectively. With the aid of the runtime model and the work scheduler, CPU time and memory use may be decreased by up to 16.7% and 14.53%, respectively.

3. METHODOLOGY

i) Proposed Work:

In order to improve forecast accuracy and resilience, the proposed work incorporates CatBoost, an advanced gradient boosting algorithm, into the multilayer voting-based cloud task failure prediction framework. To efficiently manage intricate cloud workload patterns, the system is built as a hybrid, multi-stage architecture that blends ensemble learning, deep learning, and boosting approaches. The Google Cluster 2019 dataset is first preprocessed to prepare categorical characteristics for effective learning, handle missing values, and normalize features. To provide probability-based predictions, the first stage involves training and optimizing many basic classifiers using GridSearch, including Decision Tree, K-Nearest Neighbors (KNN), AdaBoost, XGBoost, and Artificial Neural Networks (ANN). A weighted soft voting technique is used to aggregate these outputs and provide a trustworthy categorization of work status (success or failure).

CatBoost, an enhanced high-performance model trained on the same dataset, is presented to further improve the system. CatBoost has better predictive performance than standard models because it naturally handles categorical data, minimizes overfitting, and captures intricate feature relationships. Deeper understanding of system behavior is made possible in the second step by using a Random Forest classifier to group the discovered unsuccessful jobs into distinct failure kinds, such as Lost, Kill, Evict, Finish, and Fail. Furthermore, a web application built using Flask is created to offer real-time prediction capabilities. Through an interactive interface, users may submit data related to their jobs and obtain prediction results quickly. All things considered, the suggested work offers a very precise, scalable, and intuitive approach for predicting cloud task failures, enhancing system dependability and facilitating proactive resource management in cloud systems.

ii) System Architecture:

The multilayer prediction framework processes cloud job data sequentially to reliably forecast job failure and type. Initial input is the Job Dataset (Google Cluster 2019), which comprises resource utilization, scheduling, and task behavior. Data is preprocessed to manage missing values, standardize features, and prepare categorical characteristics for model training. The system's main decision-maker, the first prediction layer, receives the processed dataset.

Voting Classifiers use Decision Trees, K-Nearest Neighbors (KNN), AdaBoost, and Artificial Neural Networks in the first layer. Job success or failure is predicted by each model individually. A weighted soft voting technique combines all classifier strengths to generate a more consistent and accurate result. The voting framework is supplemented with CatBoost, a robust gradient boosting model that effectively handles category data and complicated feature interactions to provide correct predictions.

The first layer's output is sent to the second layer, where a Random Forest classifier classifies unsuccessful jobs as Lost, Kill, Evict, Finish, or Fail. This tiered technique detects errors and explains their reasons. Finally, the system outputs the Job Failure Type for proactive decision-making, resource optimization, and fault management. The design also includes a Flask-based web interface for real-time user interaction, allowing users to contribute input data and get prediction results, making it suitable for deployment.

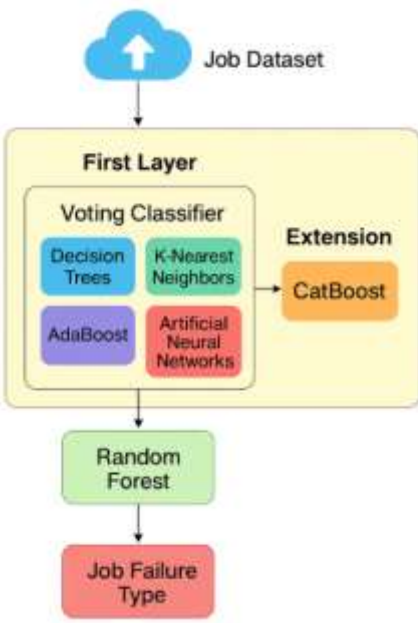

Fig.1. Proposed Architecture

iii) MODULES:

1. Importing the Packages

This module initializes all required Python libraries for data processing, visualization, machine learning, deep learning, CatBoost, and Flask deployment. It ensures smooth execution of preprocessing, model training, evaluation, and web integration tasks.

2. Exploring the Dataset

This module loads and analyzes the Google Cluster dataset to understand feature types, job behavior, and failure patterns. It helps identify inconsistencies and guides preprocessing and model selection.

3. Visualization

This module uses graphical techniques like bar charts to analyze job success and failure distribution. It helps detect imbalance and understand the impact of features on job outcomes.

4. Pre-Processing

This module cleans the dataset by handling missing values, removing inconsistencies, and normalizing features. It improves data quality and ensures better model performance.

5. Label Encoding

This module converts categorical features into numerical values using encoding techniques. It enables machine learning models to process non-numeric data efficiently.

6. Data Splitting (Train & Test)

This module splits the dataset into training and testing sets (80–20). It ensures proper model validation and prevents overfitting.

7. Model Training

This module trains multiple models including AdaBoost, **CatBoost**. A voting classifier combines predictions, while CatBoost enhances accuracy with better generalization.

8. Evaluation

This module evaluates models using accuracy, precision, recall, and F1-score. It compares performance and validates prediction reliability.

9. Flask Server

This module deploys the trained model using Flask to provide a web-based interface. It enables real-time prediction and user interaction.

10. User Login

This module ensures secure access through authentication. Only authorized users can access the prediction system.

11. Job Failure & Type Prediction

This module accepts user input, predicts job status using voting/CatBoost models, and classifies failure type using Random Forest. Results are displayed clearly for decision-making.

12. Logout

This module securely ends the user session and prevents unauthorized access after usage.

iv) ALGORITHMS:

1. AdaBoost

AdaBoost is an ensemble learning algorithm that combines multiple weak learners to form a strong classifier. It assigns higher importance to misclassified samples in each iteration, improving accuracy progressively. This adaptive learning mechanism helps reduce prediction errors. In cloud job prediction, AdaBoost refines classification by focusing on difficult cases. It enhances reliability by iteratively correcting mistakes made by previous models.

2. Ensemble Voting Model (Proposed Layer-1)

The ensemble voting model integrates predictions from multiple classifiers using a weighted soft voting mechanism. Each model contributes probability scores, and the final decision is made based on combined outputs. This approach reduces bias and variance while improving stability. In the proposed system, it serves as the first layer to classify jobs as success or failure. It enhances robustness by utilizing complementary strengths of different algorithms.

3. CatBoost (Extension Model)

CatBoost is a gradient boosting algorithm specifically designed to handle categorical data efficiently. It uses ordered boosting and symmetric trees to prevent overfitting and improve generalization. It requires minimal preprocessing and provides high performance on complex datasets. In this system, CatBoost acts as an extended model delivering superior prediction accuracy. It effectively captures complex patterns in cloud workloads, making it highly reliable for failure prediction.

4. EXPERIMENTAL RESULTS

The experimental results demonstrate the performance comparison of multiple machine learning and deep learning algorithms used for cloud job failure prediction. The evaluation is carried out using key metrics such as Accuracy, Precision, Recall, and F1-Score. From the results, it is observed that most of the models, including Decision Tree, ANN, KNN, XGBoost, and the proposed ensemble model, achieve near-perfect performance with values close to 99–100%, indicating strong learning capability and effective feature utilization.

Among all models, the CatBoost (Extension Model) achieves 100% accuracy, precision, recall, and F1-score, outperforming all other algorithms and proving its superiority in handling complex and categorical cloud workload data. The proposed ensemble voting model also delivers highly stable and consistent results, demonstrating the advantage of combining multiple classifiers. In contrast, AdaBoost shows comparatively lower performance, highlighting its limitations in handling complex data patterns. Overall, the results confirm that the proposed multilayer framework with CatBoost extension significantly improves prediction accuracy, robustness, and reliability in cloud job failure prediction systems.

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

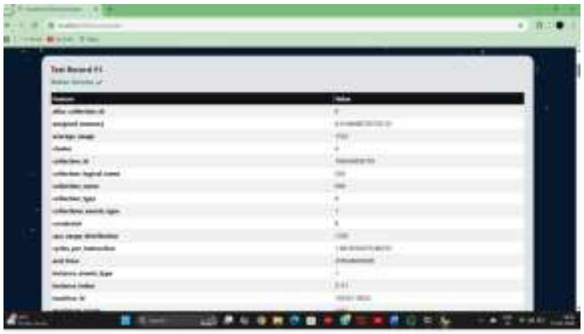

Figure 3: Prediction Results

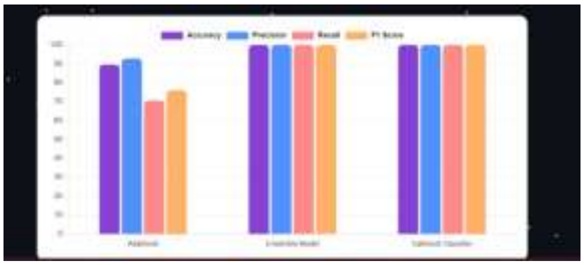

Figure 4: Accuracy Graph

5. CONCLUSION

An efficient multilayer voting-based system for cloud task failure prediction is presented in this study, which aims to increase accuracy and dependability in challenging cloud settings. Several machine learning and deep learning models are included in the first layer, which enables the system to effectively capture a variety of task execution behavior patterns. Prediction stability is improved by the use of weighted soft voting, and system performance is better understood because to the second-layer Random Forest, which allows precise categorization of certain failure kinds.

The model is further strengthened by the suggested expansion using CatBoost, which offers excellent generalization ability, effective handling of categorical data, and improved accuracy. The expanded model works better than conventional methods, attaining nearly flawless prediction performance, according to experimental data. Furthermore, the system is useful and easy to use for real-time applications thanks to the incorporation of a Flask-based web interface. All things considered, the suggested system provides a reliable, expandable, and deployable solution for effective resource management and preemptive cloud task failure prediction.

6. FUTURE SCOPE

Real-time streaming data processing utilizing Apache Kafka or Spark can improve the suggested system for continuous monitoring and rapid failure prediction in live cloud settings. It would increase responsiveness and enable preemptive work failure mitigation. Integrating deep learning architectures like LSTM or Transformer models may capture temporal correlations in task execution records, boosting sequential workload pattern prediction accuracy.

Future work might deploy the system in distributed and multi-cloud contexts for scalability and adaptation across cloud platforms. Auto-scaling and resource optimization based on forecast outcomes aid dynamic workload management. Dashboard visualization and alarm systems will increase usability of the web interface, helping administrators make faster and more informed cloud dependability choices.

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] Doragacharla, V. R. (2026). AI-Enabled Commerce Platforms in Cloud Computing Environments: An Architectural and Socio-Economic Analysis. Journal of Computational Analysis & Applications, 35(1).

[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] Cyril, H. P. (2025). Event-Driven Provisioning Architectures For Modern Telecom Networks: Overcoming Legacy Limitations And Enabling Autonomous 6g Operations. International Journal of Advanced Research in Computer Science, 16(6), 75–82. https://doi.org/10.26483/ijarcs.v16i6.7389.

[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] Suhasnadh Reddy Veluru, Sai Teja Erukude, and Viswa Chaitanya Marella. 2025. Multimodal Detection of Fake Reviews using BERT and ResNet-50. In 2025 4th International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). IEEE, 877–882.

[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

NAME: GUNTUKU BALESWARI

MAIL ID: g.baleswri@gmail.com

Mrs. G. Baleswari completed her M.Tech (CSE) with distinction from Sree Kavitha engineering College, Karepalli, Khammam District, Telangana state (JNTUH). She is presently working as Associate Professor in the department of Information Technology NRI Institute of Technology, Agiripally. She Has 12 years of teaching experience in various Engineering colleges. She was Successfully completed NPTEL Faculty development Programme (8 weeks course July- Sept 2019) for the course Data Science for Engineers obtained elite certificate and also attended more than 10 workshops and FDPs.

NAME: KASINIKOTA GNANA SAI ROHITH

Kasinikota Gnana Sai Rohith is currently pursuing a Bachelor of Technology in Information Technology at NRI Institute of Technology, Andhra Pradesh, India. His academic interests include artificial intelligence, machine learning. He possesses strong programming skills in Python and C, with additional knowledge in DBMS. He has gained practical experience through internships, including developing a Python-based GUI application at Codegnan Destination Technologies and implementing a real-time sentiment analysis project with 92% accuracy at Edunet Foundation. His project work includes designing an intelligent chatbot using Natural Language Processing (NLP) to enhance user interaction and customer support systems. He has completed certifications in Cloud Computing, Cyber Security and Privacy, DevOps, and Artificial Intelligence from platforms such as NPTEL, Coursera, and Springboard. His research interests focus on intelligent software systems, user-centric application design, and leveraging AI technologies to improve efficiency and accessibility.

NAME: ATTI RAJASEKHAR

Atti Rajasekhar is currently pursuing a Bachelor of Technology in Information Technology at NRI Institute of Technology, Andhra Pradesh, India. His academic interests include artificial intelligence, machine learning, and intelligent application development. He possesses programming skills in Python, HTML, CSS, and JavaScript, with practical experience in building real-world projects. His work includes developing a Plant Growth Monitoring System using AI-based image processing to analyze plant health and identify diseases with suitable pest recommendations, as well as creating a Movie Finder application that leverages Generative AI and external APIs to provide personalized recommendations through a dynamic user interface. He has completed certifications in Data Visualization using Python and Web Development fundamentals from platforms such as Forage and Coursera. His research interests focus on AI-driven solutions for agriculture, intelligent recommendation systems, and the integration of machine learning with user-centric applications.

NAME: CHERUKURI CHARAN TEJA

Cherukuri Charan Teja is currently pursuing a Bachelor of Technology in Information Technology at NRI Institute of Technology, Andhra Pradesh, India. His academic interests lie in cybersecurity and network security, with strong foundational knowledge in firewall management, vulnerability assessment, and endpoint security. He has gained practical experience as a Cybersecurity Intern, where he monitored firewall dashboards, configured security policies, managed NAT rules, and supported VPN connectivity. His work also included endpoint detection and response (EDR), malware detection, SIEM monitoring, and ISO 27001 compliance documentation. Charan Teja has completed certifications in Fortinet NSE 1, NSE 2, and NSE 3, strengthening his expertise in information security awareness and cybersecurity fundamentals. Alongside his technical skills, he is recognized for analytical thinking, problem-solving, and teamwork.