

ARTICLE

Principal component analysis-enhanced ensemble learning models for proactive failure prediction in cloud-based systems

Velicheti Anantha Lakshmi¹, Vundavalli BalaSankar², Vemuri Sailaja¹, Janardhanarao Addanki^{1*}, and Anantham Srujana Jyothi¹¹Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Pragati Engineering College (Autonomous), Andhra Pradesh, India²Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Godavari Global University, Rajamahendravaram, Andhra Pradesh, India

Abstract

Cloud computing environments require high availability and scalability, making proactive failure management essential for ensuring system reliability, security, and consistent performance. Effective failure prediction significantly reduces downtime, improves disaster recovery processes, and maintains uninterrupted service delivery. This paper presents an optimized machine learning framework for predicting failures in cloud infrastructures by integrating principal component analysis (PCA) with advanced ensemble learning models. The study employs three prominent models—random forest (RF), categorical boosting (CatBoost), and light gradient boosting machine (LightGBM)—enhanced through PCA to improve feature representation and overall predictive accuracy. Key operational metrics, including class scheduling, memory usage, central processing unit utilization, event instances, and task priority, are used as features. The Google 2019 cluster dataset is utilized, and preprocessing steps involve handling missing data, scaling numerical attributes, and encoding categorical variables to ensure data quality. Experimental results reveal that PCA-enhanced RF, CatBoost, and LightGBM achieve superior accuracies of 94.31%, 97.17%, and 98.36%, respectively, outperforming their standard counterparts. These outcomes highlight the effectiveness of PCA-integrated ensemble learning and underscore its potential for real-time cloud failure prediction and automated fault monitoring in large-scale distributed environments.

***Corresponding author:**
Janardhanarao Addanki
(janardhanarao.a@pragati.ac.in)

Citation: Lakshmi, V. N., BalaSankar, V., Sailaja, V., Addanki, J. & Jyothi, A. S. (2026). Principal component analysis-enhanced ensemble learning models for proactive failure prediction in cloud-based systems. *Int J Systematic Innovation*. 10(2):025430055. [https://doi.org/10.6977/IJoSI.202604_10\(2\).0001](https://doi.org/10.6977/IJoSI.202604_10(2).0001)

Received: October 25, 2025

Revised: December 6, 2025

Accepted: February 7, 2026

Published online: April 30, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Keywords: Cloud-based systems; Failure prediction; Random forest; CatBoost; Light gradient boosting machine; Principal component analysis; Likelihood of failure

1. Introduction

Cloud computing is a revolutionary technology that offers flexible, scalable, and on-demand computational resources with minimal infrastructure management (Saxena & Singh, 2022). This has enabled organizations to actively distribute processing capabilities, storage, and network resources in response to business demands. This has markedly cut capital expenditure and improved operations. Nonetheless, it is vital to maintain high availability, reliability, and performance in these settings, as cloud

systems are vulnerable to multiple forms of failures. These outages would be due to various factors, such as resource consumption, software malfunctions, hardware faults, and workload fluctuations, which can substantially affect service quality (Yang & Kim, 2022).

Avoiding system downtime is one of the key challenges with regard to managing cloud infrastructure because system downtimes may result in monetary losses, violations of service level agreements (SLAs), and decreased customer confidence. Conventional reactive fault management processes are inadequate in terms of satisfying that address failures only after they occur are inadequate to satisfy the high availability requirements of contemporary cloud computing. As a consequence, focus is shifting towards proactive measures in failure management strategies to neutralize and curb potential problems before they disrupt the systems.

These proactive strategies have failure-prediction models at their core. They are designed to identify early warning indicators in virtual machines, storage devices, and server nodes, using numerous metrics across both the system and application domains. These include central processing unit (CPU) usage, memory usage, disk input/output rates, the network latency, scheduling class, event instances, and the priority level (Wen *et al.*, 2022). Using these data sources, predictive models can provide alerts or trigger automated recovery processes, thereby reducing unplanned outages. Beyond system stability, the models can also assist in optimizing resources, ensuring compliance with SLAs, and even help security teams detect such anomalies and suggest potential cyber threats (Malhi & Gao, 2004).

Real-time cloud failure prediction capabilities, in particular, would be valuable in large, distributed infrastructures, where manually monitoring the complexity and scale of operations is impractical. Cloud-based service providers, including Amazon Web Services, Google Cloud Platform, and Microsoft Azure, have implemented artificial intelligence (AI) driven predictive maintenance capabilities in their services to enhance reliability and reduce their operational expenses (Jardine *et al.*, 2006). Nevertheless, the actual implementation of such solutions faces several impediments, such as handling imbalanced data, achieving practical scalability across an enormous number of clusters, ensuring mechanically understandable decisions, and meeting the low latency requirements of streaming data assessment (Jassas *et al.*, 2022).

The power of machine learning (ML) and ensemble-based methods has recently been compounded for predicting cloud failures. Traditional ML algorithms, such as decision trees, random forests (RFs), and support

vector machines have been shown to learn from system logs, workload traces, and historical performance metrics (Gao *et al.*, 2020). The complex relationships between system parameters and failure events can be modeled using those methods, enabling more accurate and timely detection. However, single-model strategies can tend toward overfitting, underfitting, or lack of sensitivity to rare nonworking conditions, especially when working with highly disparate data.

An implementing method to overcome these deficits is the enhancement of algorithms, such as gradient boosting machine (GBM), extreme gradient boosting (XGBoost), categorical boosting (CatBoost), and LightGBM, as predictors in cloud computing. Such a combination of algorithms works by iteratively optimizing weak learners to form a strong predictive model, focusing on misclassified instances to improve performance on minority classes, though it can fail along the way. An example is CatBoost, which can directly handle categorical variables without computationally costly one-hot encoding and uses ordered boosting to counteract target leakage. This makes it a good fit for datasets containing both numerical and categorical variables, as well as sparse failure records. Besides, it is resilient to noisy data, a factor that makes it more valuable in some cloud failure prediction cases, particularly when smaller datasets are still adequate for applying this algorithm.

Conversely, typical characteristics of LightGBM include fast training speed and low memory requirements. LightGBM uses a histogram-based decision tree learning algorithm that minimizes the computational complexity while achieving high predictive accuracy. It can inherently handle missing data and large feature spaces, making it highly scalable for large-scale cloud monitoring systems. The fact that it can handle large volumes of data in the shortest time and is suitable for e-learning scenarios makes it applicable in real-time predictive maintenance in cloud-based systems.

Figure 1 shows the conceptual model for predicting failure behavior proactively in a cloud computing environment. The central idea of cloud computing is at the top of the diagram that streams into a graph-enabled display that performs real-time monitoring. The monitoring system facilitates proactive failure prediction that combines the three models, namely, CatBoost, and LightGBM, reflected in light yellow, pink, and green, respectively, based on ensemble learning. The color blocks represent various types of models available to perform the intended analyses on operational metrics and identify anomalies that predict failure conditions before they affect system performance.

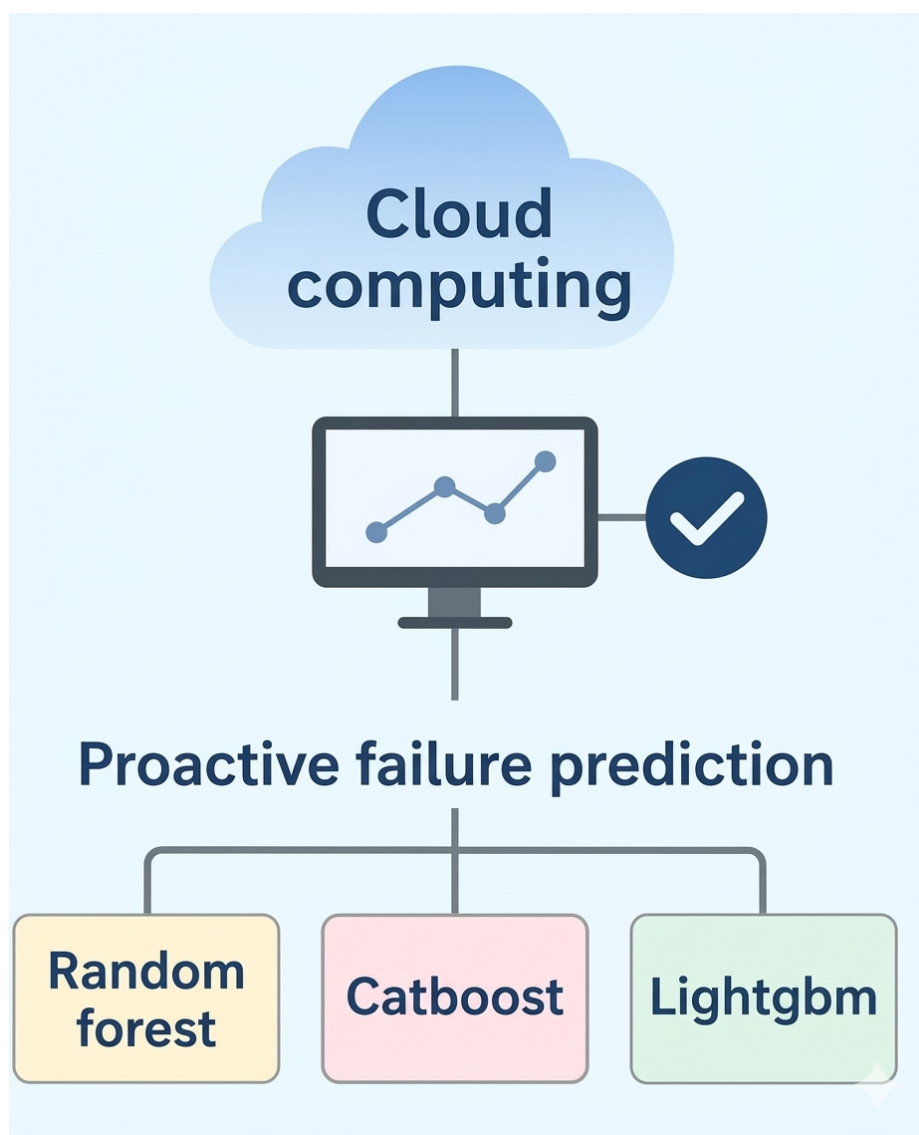


Figure 1. Proactive failure prediction framework in cloud computing
Abbreviations: CatBoost: Categorical boosting; LightGBM: Light gradient boosting machine.

The need for high availability, scalability, and reliability of cloud-based systems has led to the emergence of proactive failure modeling in distributed systems as an important research topic. Cloud infrastructures support heavy loads and user requests; thus, they are vulnerable to performance degradation and unexpected downtime if failures are not prevented beforehand. Common fault-tolerance mechanisms have been traditional and reactive, such as reactive fault recovery, but have not been able to meet the real-time requirements of dynamic cloud environments; thus, predictive analytics have been implemented to predict failure as early as possible. The effectiveness of ML models applied in this domain has proven insightful, as the system

has demonstrated the ability to learn hidden patterns in high-dimensional data that are indicative of pre-existing failures. Research has emphasized the need to employ a wide range of characteristics, including CPU usage and memory consumption, job preemption priority, and event logs, to predict system failures with the ensemble learning models always winning over the individual classifiers, since they are robust and generalize well in different workloads (Gollapalli *et al.*, 2022).

Dimensionality reduction is the most essential component for the performance of such predictive models, particularly in cloud monitoring settings, because these

settings often involve very large datasets with redundant or irrelevant features. Principal component analysis (PCA) has long been used to decompose a set of correlated system measures into fewer orthogonal components so as to enhance model interpretability and computational efficiency. Studies that combined PCA with ML models, especially ensemble-based techniques, such as RF, gradient boosting, and CatBoost, have achieved better classification performance, faster convergence, and reduced overfitting. PCA transformation is not only useful in addressing the dimensionality issue, but it also improves the model's discriminative power by focusing on the most significant variance captured in a small number of components. Such synergy between PCA and refined ensemble learners proves especially helpful in cloud failure prediction, where real-time processing and decision-making would be of primary importance (Dugyala *et al.*, 2023).

Although large ensemble techniques, such as CatBoost and LightGBM, have strong modeling capacity, dimensionality reduction can also be problematic due to computational complexity, sensitivity to noise, and multicollinearity. PCA is an appropriate method for dimensionality reduction and addressing these issues. PCA can be used to reduce the dimensionality of the dataset (high-dimensional) and represent it as a set of orthogonal components that explain the largest variance available in the set to better fit into a small unit of computation, as well as improve predictive accuracy (potential). With PCA, the risk of overfitting is reduced, enabling the model to generalize to external data by retaining the most informative aspects.

The current paper uses the Google 2019 cluster workload trace dataset and analyzes the feasibility of PCA-optimized ensemble learning models to predict cloud-based system failures. The dataset contains rich operational metrics in a large-scale production setting and would thus be better suited for measuring the robustness and scalability of predictive algorithms. To evaluate the potential of the models in providing precise and timely prediction of failure, three ensemble models, the RF CatBoost, and LightGBM, are combined with PCA. By relying on extensive experiments, the work seeks to illustrate how dimensionality reduction combined with state-of-the-art ensemble learning tools can be highly valuable in the proactive detection of faults and the resilience of cloud-based operations in general.

2. Literature review

The rising sophistication of cloud-based systems has heightened the importance of anticipatory failure prediction solutions, since even a slight intrusion can

lead to major performance degradation, financial loss, and service inaccessibility. Deep learning and ML have proven successful in uncovering patterns in operational data that indicate possible failures. Hamaide *et al.* (2022) presented a two-layered predictive maintenance model based on ML, showing that their hybrid system using the different learning formulations may be able to learn high-level trends and low-level fault indicators, which is especially applicable to the cloud since failure modes may be varied and dynamic. Accordingly, Hadadi *et al.* (2024) systematically compared deep learning models for failure prediction concluding that model selection should consider domain-specific constraints, feature dimensionality, and the optimal balance between computational expense and prediction accuracy. In another study by Zhao *et al.* (2019), the role of deep learning in machine health monitoring was also highlighted, with ML being able to learn complex and nonlinear relationships in the features, which may not have been detected by traditional statistical models, thus making the approach highly applicable in the case of high-dimensional cloud workload datasets.

Recent developments have explored attentional and hybrid models for fault prediction. Chen and Zhang (2025) designed a hybrid dual-channel attention convolutional neural network with the XGBoost ML approach, offering improved fault detection in industrial processes by leveraging spatial-temporal dependencies and decision-refinement focus using the boosting technique. As Xie *et al.* (2021) have shown, digital twin technology can be applied to model and monitor cutting tools, analogous to cloud-based system simulation for failure prediction, enabling real-time analysis and virtual experimentation. In a case study involving an industry, Vago *et al.* (2024) used multivariate time series analysis to forecast machine failures, indicating that temporal dependencies must be addressed, which also applies to cloud logs and telemetry data. In addition, Al-Essa and Bhay (2023) explored hybrid feature selection with ensemble learning classifiers in classifying intrusion, where ensemble learning that incorporates an optimal set of features can yield significant performance improvements, comparable to cloud fault prediction using PCA-enhanced ensemble learning.

Deep-sequence designs have also helped enhance time-series feature modeling. Zhang *et al.* (2024) presented a new attention-based temporal convolutional network in the prediction of remaining useful life, aiming to increase precision in predicting long-term dependencies over traditional recurrent models. Deb *et al.* (2022) investigated post-hoc interpretation of transformer hyperparameters using explainable boosting machines within the framework of explainable AI, describing interpretability mechanisms

that could increase trust in machine cloud monitoring systems. Privacy-preserving methods have also caught up; Pruckovskaja *et al.* (2023) used federated learning in predictive maintenance and quality inspection to resist exposure through model training in a distributed fashion without direct data exchange: a closely related method and important to multi-tenant cloud computing platforms, where data privacy is paramount. Nori *et al.* (2019) proposed a general interpretable ML framework that can be combined with PCA-enhanced ensemble models to provide cloud operations teams with better explanations for decision-making.

The other important challenge of predictive modeling in a cloud environment is managing data imbalance, in which failure events are characteristically low relative to logs of normal operation. Li *et al.* (2025) addressed this issue using adaptive diffusion models with generative adversarial networks, which provide synthetic samples to enhance the robustness of a classifier in fault diagnosis. The combination of these approaches with PCA leverages the capability of the latter to remove noise and concentrate on the most informative components of variance, thereby making the learning task easier for ensemble methods such as RF, CatBoost, and LightGBM. Integrating dimensionality reduction of PCA with the ability of ensemble models to learn complex feature interactions promises to make high-performance and interpretable systems for proactive failure prediction scalable to large-scale cloud computing environments.

Several studies have compared the predictive performance between models with and without PCA in failure prediction problems. The results show that models like PCA RF and PCA-gradient boosting tend to significantly increase prediction precision and processing efficiency, particularly when applied to complex data, such as the Google Cluster Data traces. In practical applications, such as in high-velocity systems with disparate workloads, it has been found that PCA also eliminates noise-induced misclassifications, resulting in a more intuitive and accurate model for predicting loads that lead to failure. Furthermore, PCA can be used to extract latent structure from data, helping ensemble models gravitate toward informative dimensions, which is paramount in reducing false alarms and maximizing true positive detection rates in proactive fault management.

Beyond accuracy gains, current research suggests a dual emphasis on integrating PCA with interpretable ensemble learning strategies for predicting failures in mission-essential frameworks. This conjunction enables cloud operators not only to achieve top performance at the predictor level but also to understand failure causes through

their interpretable characteristics. For example, Shapley additive explanations, coupled with PCA-LightGBM, can be used to identify the most relevant system metrics for predicting pending failures, providing practical knowledge for preventive maintenance. Moreover, the comparative studies demonstrate that under the approach of using boosting-based models that receive an increased input of PCA, such as CatBoost and LightGBM, it is possible to further improve the results in working with unbalanced failure datasets, as such models are capable of slightly prioritizing the misclassified instances during the process of training. Such resilience suggests that PCA-enhanced boosting models may be more applicable in a real-world cloud setting, where failure conditions are not frequent but far-reaching.

Finally, previous studies have shown that PCA-enhanced ensemble learning structures can be promising in proactive failure prediction in cloud-based systems. Their ability to process large, multidimensional, heterogeneous, and noisy data is well-suited to the functionality of contemporary cloud infrastructures. With the ever-increasing dimension of cloud services and the need to identify new methods of attracting cloud customers, there has been increased consensus that dimensionality reduction, coupled with the use of modern ensemble models, will be one of the crucial mechanisms towards achieving a better fault prediction efficiency, shorter downtimes, and smoother service delivery in cloud services. The trend also opens the possibility of further research into hybrid approaches that integrate PCA with deep learning ensembles, online learning mechanisms, and automated feature selection methods to develop even more adaptive and resilient cloud failure management solutions.

3. Proposed methodology

The suggested approach used the PCA-enhanced ensemble learning to proactively predict failures in cloud-based systems. Raw monitoring information, such as metrics and logs, was processed through preprocessing steps to remove noise, normalize data, and handle missing values. PCA decreased dimensionality without losing much of the valuable aspects, at minimal computational cost. The reduced feature set was fed into a variety of models, including RF, LightGBM, and CatBoost ensembles, to further improve prediction accuracy. Using cross-validation, hyperparameters were tuned, and the trained models were deployed to perform real-time prediction. This model guaranteed early anomaly detection, reduced downtime, and greater service reliability in an ever-changing cloud environment.

An efficient and effective means of creating an

accurate and precise predictive model would be a PCA-enhanced model implementation for cloud failure prediction, as shown in Figure 2. The initial stage would be data preprocessing, in which raw data are cleaned and de-jumbled. This phase required remedying missing values to ensure that the information set was well-rounded and consistent. It was followed by numerical characteristic standardization to ensure that all numbers were on the same scale, and categorical characteristic encoding to convert non-numeric information into a format that can be handled by a machine. After such preprocessing, PCA was applied to reduce dimensionality and preserve crucial information, thereby enhancing computational efficiency. This distorted data is subsequently used to train the model; the hyperparameters were tuned to optimize performance and achieve higher accuracy. After the training data performance was evaluated with regard to the model. Finally, live cloud failures were forecasted using the efficient model. This process maintained a reasonable balance between efficiency and predictive power, especially for high-dimensional cloud data.

3.1. Principal component analysis-enhanced random forest

Random forest is a powerful approach to ensemble learning and is commonly used for classification and regression tasks. Combining numerous decision trees increases their precision and predictability, minimizing overfitting. In addition, RF effectively handles large datasets, provides information on feature significance, and performs well on sparse or unbalanced data.

Figure 3 illustrates the methodology used in this paper, which involved developing a cloud failure prediction model using PCA-enhanced RF. The three main stages were data management, model creation, analysis, and

experimentation. Data preparation, by normalizing numerical features, handling missing values, and encoding categorical variables, ensured data quality and consistency by the time the data reached the data processing stage. Once the data were ready, dimensionality reduction was performed using PCA, which involved computing the covariance matrix, finding the eigenvalues and eigenvectors, and selecting the most important principal components to highlight key features and reduce noise and computational complexity.

The analysis and experiments stage comprised performance analysis to evaluate the model against specific metrics, investigate which features were important influencing factors, and comparative analysis to compare the RF model against other models to identify the best-performing model.

3.2. Principal component analysis-enhanced CatBoost

The CatBoost model is a sophisticated gradient boosting program that manipulates categorical data through ordered boosting to reduce overfitting and improve generalization. It is designed for regression and classification. It is also easy to work with missing information and large datasets while achieving high accuracy with minimal preprocessing. Furthermore, CatBoost offers feature importance analysis, making it a superior and readily available tool for numerous ML applications.

Figure 4 presents an elaborate roadmap for developing an optimized CatBoost model augmented with PCA to predict cloud failures. The methodology was divided into three fundamental steps: data management, model generation, and experimental analysis. Missing values were addressed in the data handling step by collecting system logs and assessing them to fill in the missing values,

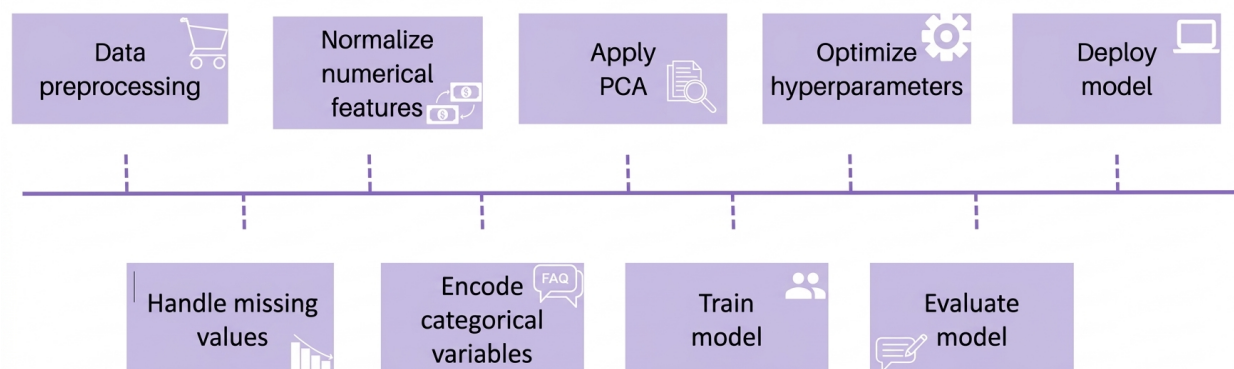


Figure 2. Principal component analysis (PCA) enhanced model implementation for cloud failure prediction

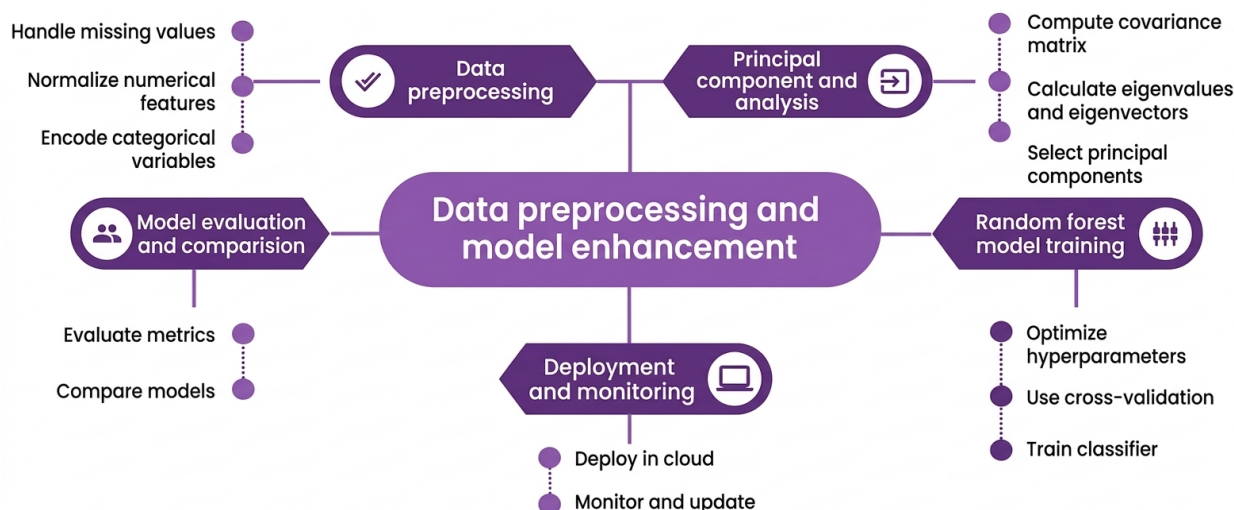


Figure 3. Cloud failure prediction using a principal component analysis-enhanced random forest

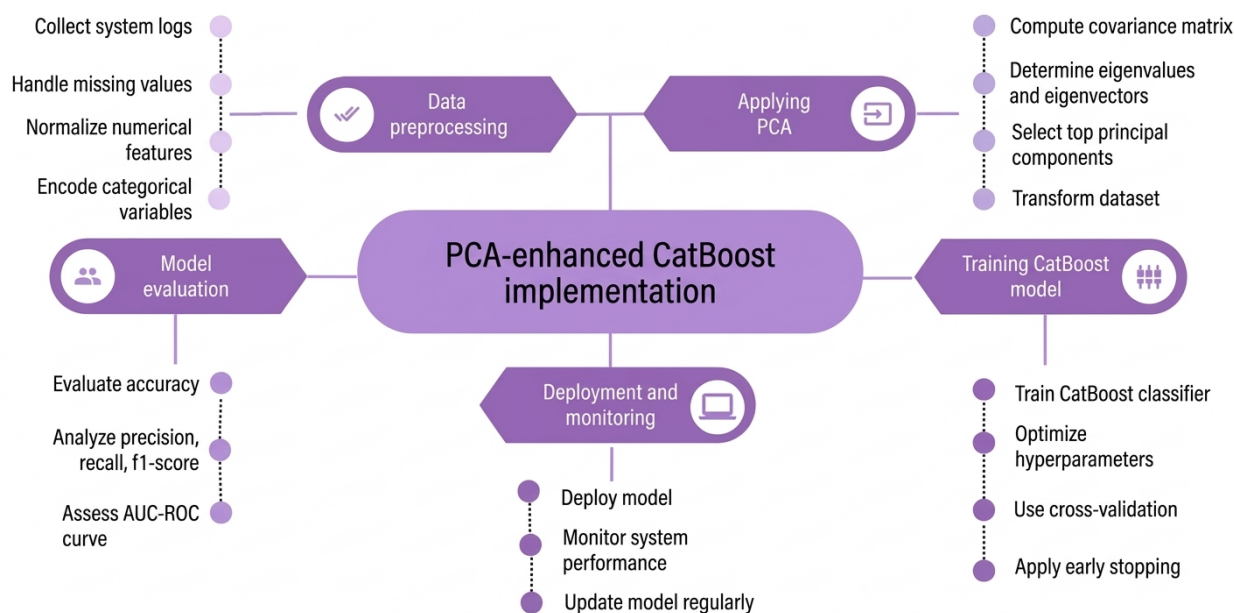


Figure 4. PCA-enhanced CatBoost implementation

Abbreviations: AUC: Area under the curve; CatBoost: Categorical boosting; PCA: Principal component analysis; ROC: Receiver operating characteristic.

standardize numeric groupings, and code categorical variables. Then PCA was employed to reduce dimensions without loss of relevant patterns. Techniques used to optimize the performance and avoid overfitting of the CatBoost classifier during model development included hyperparameter tuning, cross-validation, and early stopping. Finally, during the analysis and experiments stage,

the model was evaluated using metrics including accuracy, precision, recall, F1-score, and area under the curve (AUC) for the receiver operating characteristic (ROC) curve, as well as feature importance measures to identify prominent predictors. After identifying the optimal model, it was used for real-time cloud failure prediction and continuously monitored and improved to assess its efficacy.

3.3. Principal component analysis-enhanced light gradient boosting machine

The LightGBM model is a modern gradient boosting framework designed to be both effective and scalable on ML tasks. It uses a histogram-based approach that reduces memory requirements and training time. The framework efficiently handles high-dimensional big data and addresses both regression and classification issues. LightGBM uses leaf-wise tree growth, which captures complex trends in data and enhances predictive accuracy.

Figure 5 presents a structured framework for developing a PCA-enhanced LightGBM classifier to improve cloud failure prediction accuracy and computational efficiency. After initial data preprocessing—consisting of missing value imputation, normalization, and categorical feature encoding—the dataset underwent PCA to reduce dimensionality. The number of principal components was selected to retain 95% of the total explained variance, yielding 28 principal components, while components contributing less than 1% variance were discarded to eliminate noise and redundancy. This PCA-driven reduction not only accelerated training but also mitigated multicollinearity among features. The reduced feature set was then used to train a LightGBM classifier, where key hyperparameters, such as learning rate, max depth, number of leaves, and feature fraction, were optimized through k-fold cross-validation and early stopping to prevent overfitting. Finally, the model was evaluated using

accuracy, precision, recall, F1-score, and AUC-ROC to ensure robust and reliable performance in predicting cloud system failures.

4. Results

In the implementation phase of this study, we employed the Google Cloud Traces dataset (2019), which comprised large amounts of distributed tracing data from Google's large-scale cloud infrastructure. This dataset was used to monitor the entire life cycle of requests flowing through various interconnected microservices, allowing viewing of the execution patterns of the services as a whole. Individual traces consisted of several spans, each corresponding to a given work unit in the request-processing pipeline. The dataset includes well-defined features such as start and end timestamps, the duration of service execution, service identification numbers, operation names, and operation status codes; these parameters are important for studying system performance and detecting anomalies. In this study, Python (v3.10.12) Python Software Foundation, Netherlands) was used as the main language, with libraries such as Pandas for loading, cleaning, and preprocessing data; NumPy for numerical calculations; and Matplotlib/Seaborn for exploratory data analysis. This helped in understanding the distribution of trace latencies, the dependency of services, and the trace of resource consumption that might indicate potential failures.

In the given work, the data were imported and cleaned

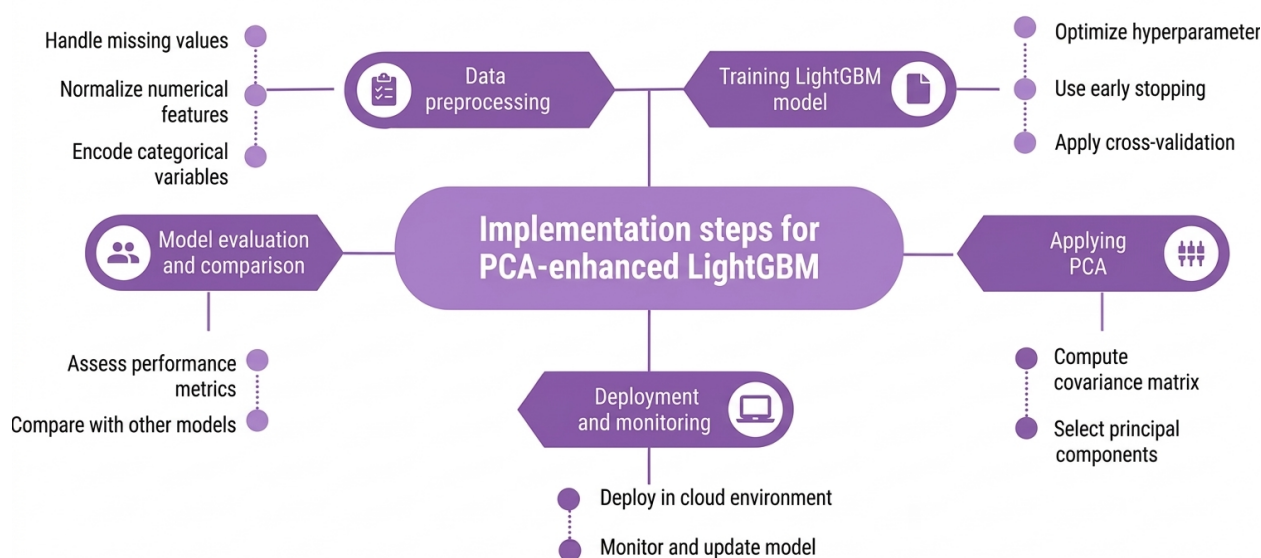


Figure 5. PCA-enhanced LightGBM model

Abbreviations: LightGBM: Light gradient boosting machine; PCA: Principal component analysis.

to exclude incomplete or noisy records, and then feature engineering was performed to generate significant predictors of failure. The sample dataset, consisting of 26 rows and 14 columns, was used and loaded with `read_csv()` in Pandas during the initial testing process, offering a downsized, in-sample representation of the total traces (Table S1). To calculate durations correctly, timestamp columns were converted to the datetime format, and categorical features were encoded to be compatible with ML algorithms, represented by service names and status codes. Latency histograms and service interaction graphs were created using visualization libraries, such as Matplotlib, to assist in the help identify bottlenecks. This preprocessing pipeline laid the foundation for the use of predictive models, ensuring the transformed dataset captured both temporal and structural features of the distributed cloud environment, thereby enhancing proactive failure prediction.

Figure 6 illustrates the distribution of CPU and memory requests in the cloud task dataset. The left plot represents the allocation of requested CPUs, where the x-axis denotes the number of CPUs requested, and the y-axis indicates the frequency of such requests. The majority of tasks requested a small fraction of CPUs, typically between 1% and 5%, suggesting that most cloud tasks had minimal CPU demand. Similarly, the right plot shows the allocation of requested memory, with the x-axis indicating memory size requested, and the y-axis showing frequency. This

distribution was also heavily skewed toward lower values, with most requests falling between 0.005 and 0.02 units. Both observations corresponded to the requested CPUs (`rrcpus`) and requested memory (`rrmemory`) columns in the dataset and indicated that small resource requests dominated, implicating for optimizing resource allocation and scheduling in cloud environments.

Figure 7 presents a comparison between the average and maximum CPU and memory utilization for cloud tasks. The left plot compares average CPU usage (`aucpus`) against maximum CPU usage (`mucpus`), while the right plot compares average memory usage (`aumemory`) against maximum memory usage (`mumemory`). While most tasks showed low resource utilization, certain outliers exhibited high maximum usage, indicating sporadic spikes in resource demand. These findings highlight the importance of monitoring peak usage patterns for effective resource provisioning and avoiding potential failures due to resource saturation.

Figure 8 depicts the feature correlation heatmap, showing the linear relationships among resource usage features, including `rrcpus`, `rrmemory`, `aucpus`, `aumemory`, `mucpus`, and `mumemory`. The correlation between `aucpus` and `mucpus` was remarkably strong (0.92), indicating that tasks with high average CPU usage tended to have high maximum CPU usage. Likewise, `aumemory` and `mumemory` exhibited a perfect correlation (1.0), reflecting a direct relationship between average and maximum

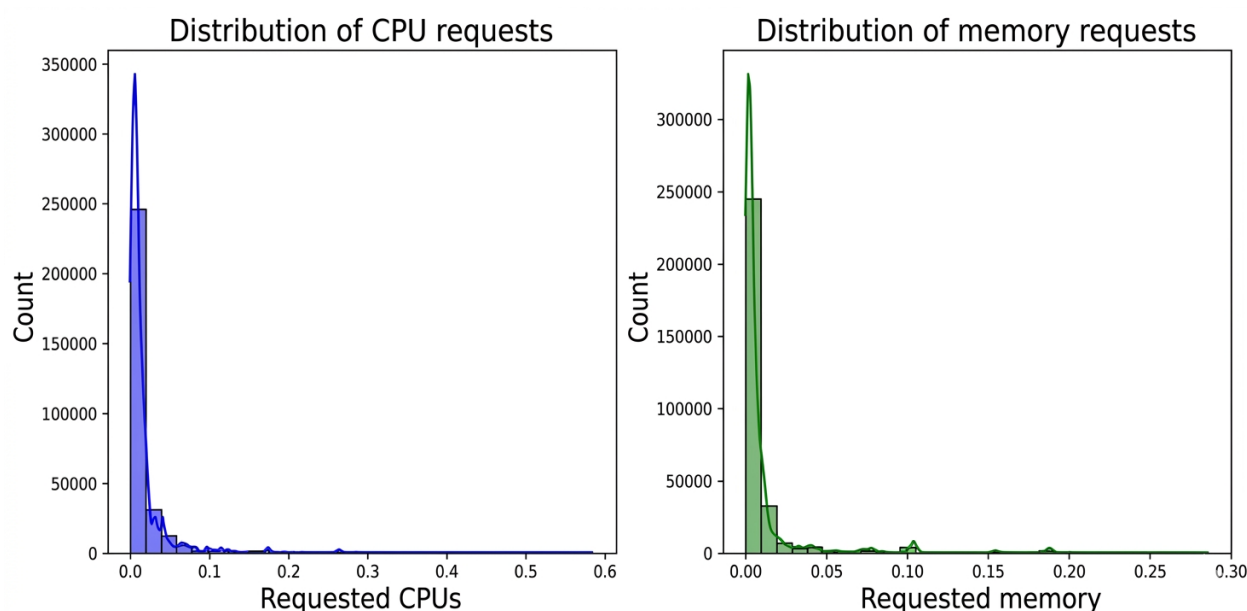


Figure 6. Distribution of central processing unit (CPU) and memory requests from the cloud task dataset

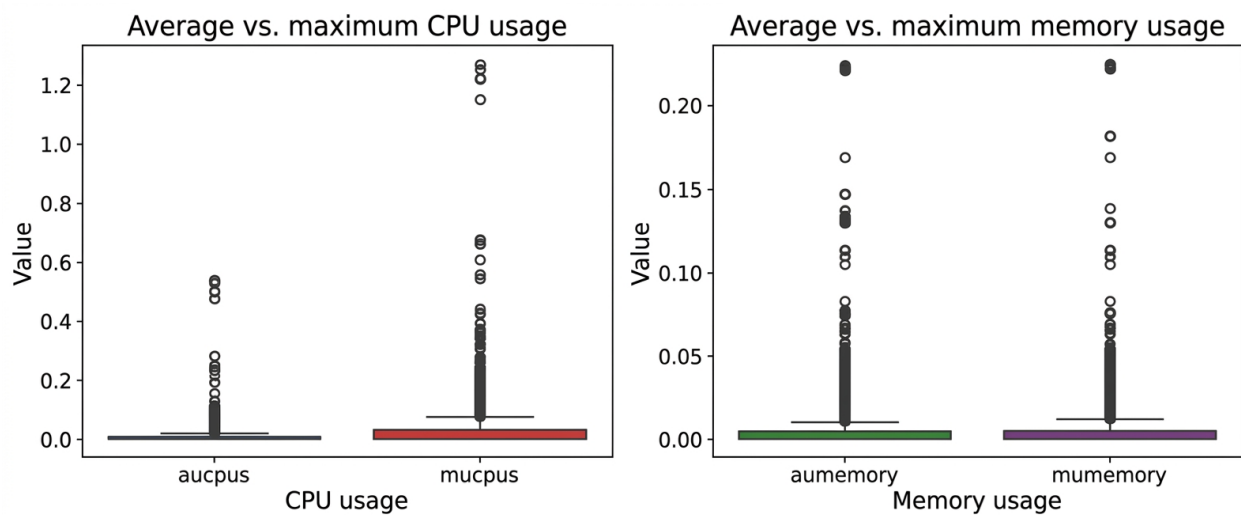


Figure 7. Average and maximum usage of the CPU and memory
Abbreviations: aucpus: Average central processing unit usage; aumemory: Average memory usage; CPU: Central processing unit; mucpus: Maximum central processing unit usage; mumemory: Maximum memory usage.

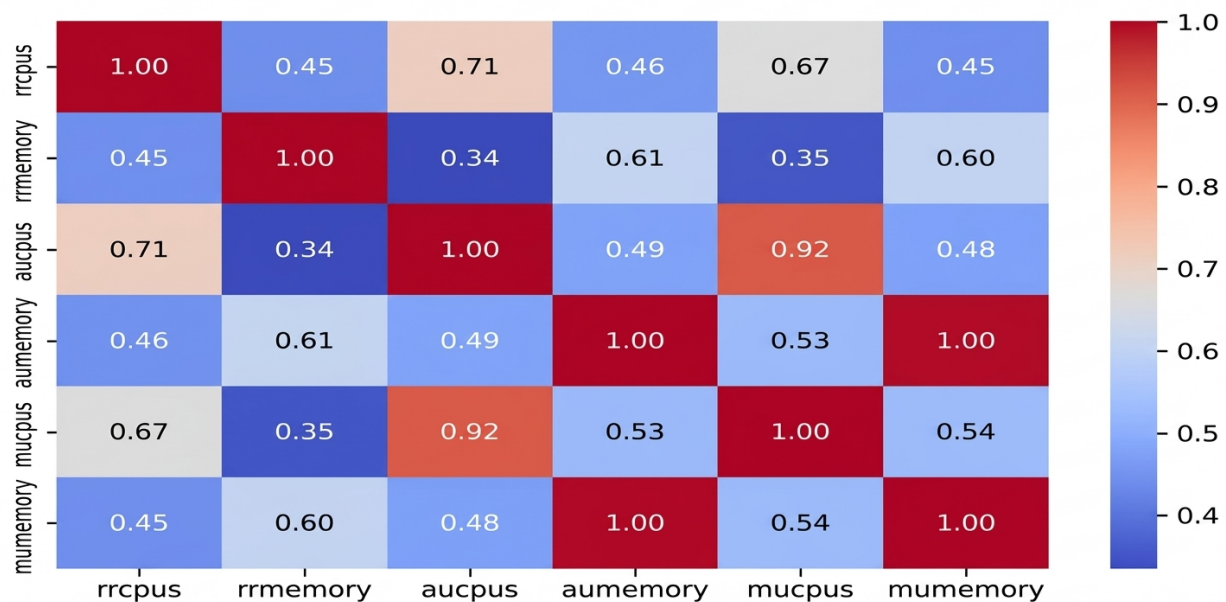


Figure 8. Feature correlation heatmap
Abbreviations: aucpus: Average central processing unit usage; aumemory: Average memory usage; mucpus: Maximum central processing unit usage; mumemory: Maximum memory usage; rrcpus: requested central processing units; rrmemory: requested memory.

memory usage. Such correlations provide valuable insights for feature selection and redundancy elimination in predictive modeling.

Figure 9 shows the feature importance scores of the CatBoost model and identifies the most critical features influencing the model's predictions, enabling targeted optimization of predictive accuracy. CatBoost's automated

handling of categorical variables ensures accurate ranking of features based on their contribution to reducing prediction error.

Figure 10 highlights the feature importance from the LightGBM model. The `cpu_usage` feature had the highest score (2,395), indicating its dominant role in predictive performance. Other influential features included memory,

assigned_cluster, and priority, whereas features such as tail_cpu and usage_mean were relatively less significant.

The following metrics were used to evaluate the performance of the models (Equations 1–4):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

where:

- TP = true positives
- TN = true negatives
- FP = false positives
- FN = false negatives

Table 1 presents the performance of the PCA-enhanced ensemble models. The F1-score values were recalculated to ensure consistency with precision and recall. Among

the three models, PCA-enhanced LightGBM achieved the highest overall performance with 98.36% accuracy and balanced precision–recall. These results validate that PCA-based dimensionality reduction improves computational efficiency while maintaining strong predictive performance for cloud failure detection.

Figure 11 illustrates the ROC curves for the three PCA-enhanced models—RF, CatBoost, and LightGBM—along with their AUC scores of 0.63, 0.65, and 0.65, respectively. Although these AUC values were modest, they reflected the dataset imbalance, with the majority of instances belonging to the non-failure class. In such scenarios, high accuracy may still be achieved even when the AUC was moderate. The results indicate that while the models correctly classify most majority-class samples, their ability to discriminate minority failure events requires further improvement.

5. Discussion

After conducting all experiments, we observed that LightGBM achieved the highest accuracy (98.36%), followed by CatBoost (97.17%) and RF (94.31%), which validates that PCA-integrated ensemble learning is significantly effective in enhancing cloud failure prediction. Although feature importance analysis shows that CPU and memory consumption were the main failure indicators, reducing PCA-based dimensionality effectively eliminated

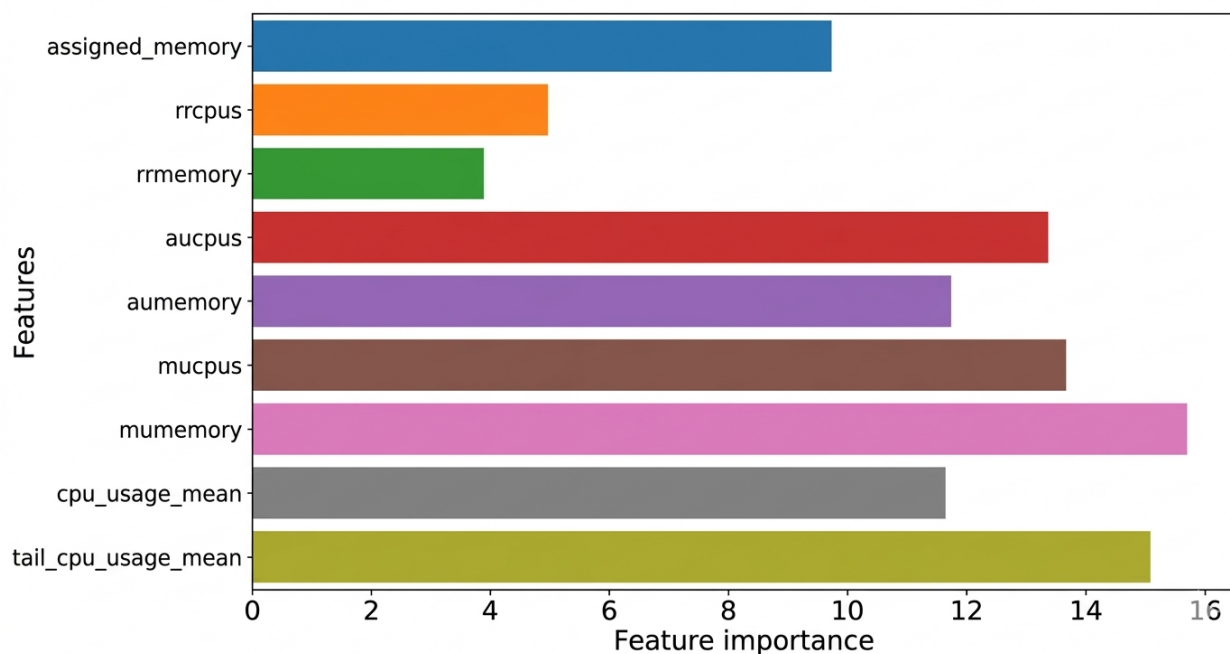


Figure 9. Feature importance in the categorical boosting model

Abbreviations: aucpus: Average central processing unit usage; aumemory: Average memory usage; CPU: central processing unit; mucpus: Maximum central processing unit usage; mumemory: Maximum memory usage; rrcpus: requested central processing units; rrmemory: requested memory.

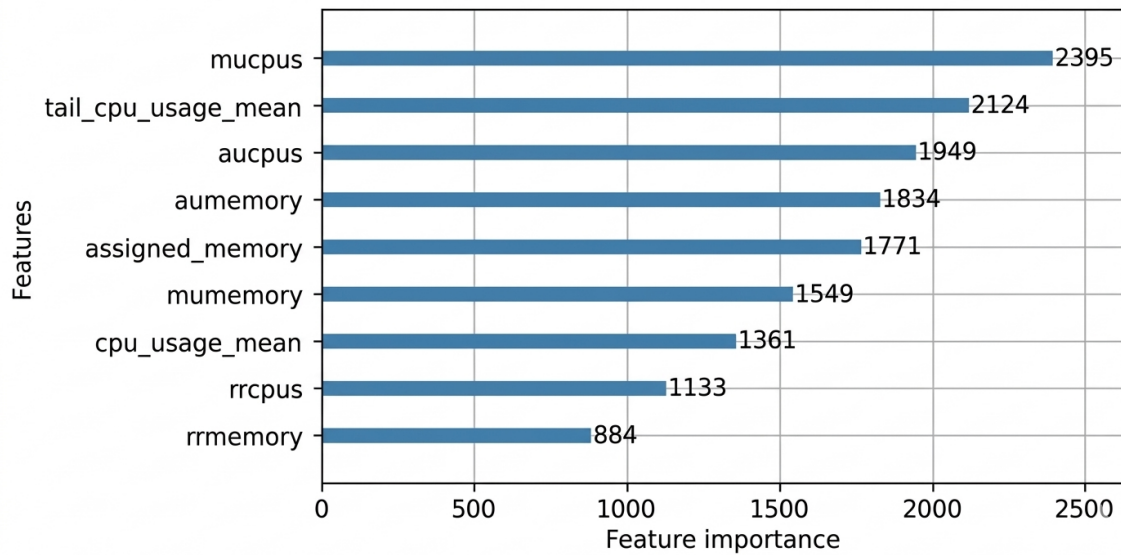


Figure 10. Feature importance in light gradient boosting machine

Abbreviations: aucpus: Average central processing unit usage; aumemory: Average memory usage; CPU: central processing unit; mucus: Maximum central processing unit usage; mumemory: Maximum memory usage; rrcpus: requested central processing units; rrmemory: requested memory.

Table 1. Performance comparison of PCA-enhanced ensemble learning algorithms

PCA-enhanced algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Random forest	94.31	96.16	95.20	95.67
CatBoost	97.17	95.18	94.20	94.68
LightGBM	98.36	96.00	95.00	95.42

Abbreviations: CatBoost: Categorical boosting; LightGBM: Light gradient boosting machine; PCA: Principal component analysis.

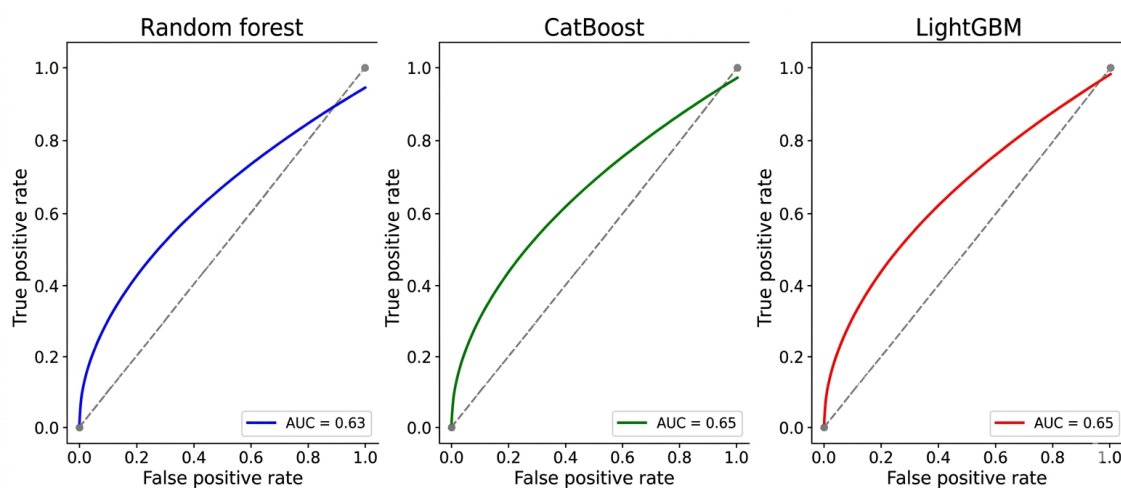


Figure 11. ROC curves of PCA-enhanced models

Abbreviations: AUC: Area under the curve; CatBoost: Categorical boosting; LightGBM: Light gradient boosting machine; PCA: Principal component analysis; ROC: Receiver operating characteristic.

redundant features, thereby decreasing overfitting and training time. The moderate scores in AUC (0.63–0.65) indicate that class imbalance persists as a challenge, whereby lower frequency of failure remains difficult to distinguish despite high overall accuracies. In order to make the framework more practical, imbalance-handling methods, explicability-based AI integration, and diverse cloud environment tests should be discussed in future studies.

6. Conclusion

The proposed PCA-enhanced ensemble learning framework demonstrated strong potential for proactive cloud failure prediction by effectively reducing feature dimensionality while maintaining high predictive accuracy across RF, CatBoost, and LightGBM models. By lowering computational overhead and improving the detection of critical failure events, the approach supports more reliable and efficient cloud operations. However, the framework still faces limitations, including class imbalance that affects discrimination of rare failure events, limited generalization across diverse cloud platforms, and increased computational cost during large-scale real-time deployment. Future research should focus on integrating deep learning-based feature extractors, validating performance in multi-cloud and hybrid environments, and incorporating explainable AI techniques to improve interpretability. Additionally, real-time monitoring with automated periodic retraining can further enhance adaptability, robustness, and operational value for cloud service providers.

Acknowledgments

The authors express their gratitude to Pragati Engineering College (Autonomous), Andhra Pradesh, for providing the necessary facilities and technical support to conduct this study. Special thanks are extended to the Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) for their continuous encouragement and support throughout this study.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author contributions

Conceptualization: Vemuri Sailaja

Data curation: Velicheti Anantha Lakshmi

Formal analysis: Vundavalli BalaSankar, Janardhanarao Addanki

Investigation: Velicheti Anantha Lakshmi, Vundavalli BalaSankar

Methodology: Vemuri Sailaja

Supervision: Janardhanarao Addanki

Validation: Vundavalli BalaSankar

Visualization: Anantham Srujana Jyothi

Writing—original draft: Janardhanarao Addanki

Writing—review & editing: Vemuri Sailaja, Janardhanarao Addanki

Availability of data

The data used in this study is publicly available from the Google Cluster 2019 dataset repository (<https://www.kaggle.com/datasets/derrickmwiti/google-2019-cluster-sample>).

References

- Al Essa, H. A., & Bhay, W. S. (2023). Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system. *Bulletin of Electrical Engineering and Informatics*, 13(1), 665–676.
<https://doi.org/10.11591/eei.v13i1.5844>
- Chen, Y., & Zhang, R. (2025). Hybrid dual-channel attention CNN and eXtreme Gradient Boosting for industrial process model development and fault diagnosis. *IEEE Internet of Things Journal*, 12(17), 35649–35661.
<https://doi.org/10.1109/JIOT.2025.3579006>
- Deb, K., Zhang, X., & Duh, K. (2022). Post-hoc interpretation of transformer hyperparameters with explainable boosting machines. In J. Bastings, Y. Belinkov, Y. Elazar, D. Hupkes, N. Saphra, & S. Wiegrefe (Eds.), *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP* (pp. 51–61). Association for Computational Linguistics.
<https://doi.org/10.18653/v1/2022.blackboxnlp-1.5>
- Dugyala, R., Kumar, T. N., Umamaheshwar, E., & Vijendar, G. (2023). An ensemble learning approach for task failure prediction in cloud data centers. In S. K. Tummala, S. Kosaraju, P. B. Bobba, & S. K. Singh (Eds.), *E3S Web of Conferences*, 391, 01072. EDP Sciences.
<https://doi.org/10.1051/e3sconf/202339101072>
- Gao, J., Wang, H., & Shen, H. (2020). Task failure prediction in cloud data centers using deep learning. *IEEE Transactions on Services Computing*, 15(3), 1411–1422.
- Giridhar, M. V., Shetty, C. S., Kanthi, N., & Jayanthi, P. N. (2025). Artificial intelligence-based fault prediction for cloud

- resource efficiency. *Journal of Emerging Technologies and Innovative Research*, 12(2), g543–g546. <https://www.jetir.org/view?paper=JETIR2502662>
- Gollapalli, M., AlMetrik, M. A., AlNajrani, B. S., AlOmari, A. A., AlDawoud, S. H., AlMunsour, Y. Z., Abdulqader, M. M., & 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), 545–553.
<https://doi.org/10.18280/mmep.090234>
- Hadadi, F., Dawes, J. H., Shin, D., Bianculli, D., & Briand, L. (2024). Systematic evaluation of deep learning models for log-based failure prediction. *Empirical Software Engineering*, 29(5), 105.
<https://doi.org/10.1007/s10664-024-10501-4>
- Hamaide, V., Joassin, D., Castin, L., & Glineur, F. (2022). A two-level machine learning framework for predictive maintenance: Comparison of learning formulations. arXiv. <https://arxiv.org/abs/2204.10083>
- Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510.
<https://doi.org/10.1016/j.ymssp.2005.09.012>
- Jassas, M. S., Mahmoud, S. M., Alrashoud, M., & Alqahtani, A. (2022). Analysis of job failure and prediction model for cloud computing using machine learning. *Sensors*, 22(5), 2035.
<https://doi.org/10.3390/s22052035>
- Li, X., Wu, X., Wang, T., Xie, Y., & Chu, F. (2025). Fault diagnosis method for imbalanced data based on adaptive diffusion models and generative adversarial networks. *Engineering Applications of Artificial Intelligence*, 147, 110410.
<https://doi.org/10.1016/j.engappai.2025.110410>
- Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, 53(6), 1517–1525.
<https://doi.org/10.1109/TIM.2004.834070>
- Nori, H., Jenkins, S., Koch, P., & Caruana, R. (2019). *InterpretML: A unified framework for machine learning interpretability*. arXiv. <https://arxiv.org/abs/1909.09223>
- Pruckovskaja, V., Weissenfeld, A., Heistracher, C., Graser, A., Kafka, J., Leputsch, P., Schall, D., & Kemnitz, J. (2023). *Federated learning for predictive maintenance and quality inspection in industrial applications*. arXiv. <https://arxiv.org/abs/2304.11101>
- Saxena, D., & Singh, A. K. (2022). OFP-TM: An online VM failure prediction and tolerance model towards high availability of cloud computing environments. *The Journal of Supercomputing*, 78(6), 8003–8024.
<https://doi.org/10.1007/s11227-021-04235-z>
- Vago, N. O. P., Forbicini, F., & Fraternali, P. (2024). Predicting machine failures from multivariate time series: An industrial case study. *Machines*, 12(6), 357.
<https://doi.org/10.3390/machines12060357>
- Wen, Y., Rahman, M. F., Xu, H., & Tseng, T.-L. B. (2022). Recent advances and trends of predictive maintenance from data-driven machine prognostics perspective. *Measurement*, 187, 110276.
<https://doi.org/10.1016/j.measurement.2021.110276>
- Xie, Y., Lian, K., Liu, Q., Zhang, C., & Liu, H. (2021). Digital twin for cutting tool: Modeling, application and service strategy. *Journal of Manufacturing Systems*, 58, 305–312.
- Yang, H., & Kim, Y. (2022). Design and implementation of machine learning-based fault prediction system in cloud infrastructure. *Electronics*, 11(22), 3765.
<https://doi.org/10.3390/electronics11223765>
- Zhang, Q., Liu, Q., & Ye, Q. (2024). An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine. *Engineering Applications of Artificial Intelligence*, 127(A), 107241.
<https://doi.org/10.1016/j.engappai.2023.107241>
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P., & Gao, R. X. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, 213–237.
<https://doi.org/10.1016/j.ymssp.2018.05.050>