International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 565–574
https://doi.org/10.70917/ijcisim-2026-2381
Article

LSTM-ANN: A Hybrid Deep Learning Model for Task Failure Prediction in Cloud Computing Environment

Mannu[1], Jai Bhagwan[1*], Seema Rani[2], Sanjeev Kumar[1], Sunila Godara[1]
[1]Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, India.
[*]Email: drjaicse@gmail.com, ORCID: 0000-0002-8708-4029
[2]Department of Computer Science & Engineering, Ch. Devi Lal State Institute of Engineering & Technology, Sirsa, India.

Abstract: A cloud computing environment can perform millions of tasks per day, and reliability and efficient use of resources are significant issues for cloud service providers. Task failure due to workload imbalance, resource contention, hardware failure, and abnormal task execution behavior can have a dramatic impact on system performance and negatively impact on SLAs. In this research, we have proposed hybrid deep learning model for task failure prediction namely LSTM-ANN. The simulation was performed in Python language and Google Cluster Traces 2019 dataset was used for performance checking. The proposed framework compared with Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), and Hybrid GRU-ANN, to perform binary classification of failed and non-failed cloud tasks. The study involves data preprocessing, feature selection using SelectKBest, ANOVA F-test scoring function, and workload-based event transformation in order to enhance the performance of the prediction. Evaluation of the proposed model was performed using accuracy, F1-score, ROC-AUC, RMSE, and confusion matrix analysis. The experimental results showed that the Hybrid LSTM-ANN model had the highest overall accuracy of 96.08%, F1 score of 0.9173, ROC-AUC value of 0.9925, and the lowest RMSE value of 0.1721. The results of the Hybrid GRU-ANN model were also very competitive in terms of performance but not accurate as the LSTM-ANN model produced.
Keywords: Artificial Neural Network, Cloud Computing, Deep Learning, GRU, LSTM, LSTM-ANN, Task Failure Prediction

1. INTRODUCTION

Cloud computing has become the basic infrastructure of information technology in the modern world, offering on-demand, scalable, and economical resources to industrial web services as well as scientific workflows. The growth of dependency on this infrastructure, however, has increased the complexity of supporting data centers, which today operate millions of operations a day. Failures in these large-scale environments are not statistical events, but rather common occurrences caused by hardware failures, software exceptions, and resource contention. Such disruptions are very harmful; for users, they lead to data loss and time wastage, whereas for Cloud Service Providers (CSPs), they result in hefty fines due to breach of Service Level Agreement (SLA) as well as the wastage of valuable computational resources, such as CPU and memory. Traditional fault tolerance mechanisms, such as reactive check pointing, are increasingly being considered inadequate because these strategies incur high latency and overhead in detecting a failure. This leads to a paradigm shift in proactive failure prediction, where systems can predict failures of various tasks beforehand and instigate processes such as dynamic task migration or resource rescaling in preparation. Although early predictive systems employed conventional machine learning, they often struggled with longer-term time dependencies and imbalanced class factors in cloud traces. Recent studies are currently shifting their focus towards Deep Learning (DL) designs, which are efficient at detecting system precursors to failure in high-dimensional telemetry and log information. The forecasting of task failures is not a simple task because of cloud workloads that are dynamic and stochastic. Tasks within a cluster of clouds have complex dependencies and extremely changeable resource consumption behaviors. Initial studies were based on statistical techniques and conventional algorithms of machine learning (ML). Such models as Random Forest (RF) and K-Nearest Neighbors (KNN) can successfully find correlations between resource use (e.g., CPU

intensity) and failure, but they tend to fail at finding long-term temporal correlations [1][2]. Moreover, there is the issue of imbalance in the classes: the number of successful tasks is so much greater than the number of failed tasks. To solve that, the solution is based on domain information mining, where similar jobs could be clustered to discover inherent structural correlations that might enhance prediction generalization even when the stage-based dynamic information is inadequate [3]. Modern forecasting of failure primarily relies on sequential data processing as the basis of its functioning. The basic approach proposed by Gao et al. [4] is the one that relies on the work of Bidirectional LSTM (Bi-LSTM) networks. Unlike regular LSTMs, Bi-LSTMs are bi-directional, i.e., they process a situation in both directions, forward and backward, in this manner exhaustively. They demonstrated that Bi-LSTM was significantly higher than the baseline RNNs on their project on Google Cluster Traces. Similarly, Salanke et al. contributed to it with a combination of Bi-LSTM and dynamic task migration. They have a migration algorithm initiated automatically in their structure, which is the realization of the imminent high chance of failure, and thus they are virtually repairing the system before they go down [5].

Rest of the paper is organized as: section 2 presents literature review, in section 3 proposed methodology is explained, results and discussion are described in section 4, finally conclusion and future scopes are discussed in section 5.

2. Literature review

Aldomi et al. (2026) concentrated on the early prediction of failures in the tasks. They claimed that it is too late in a task to predict failure and do something about it. They have used deep learning models to detect the precursors of failure in the early phases of the task execution, which enables more time-saving rescheduling of the resources [5]. The work of Wu (2025) studied the fault detection models, having a certain focus on resource optimization. It was pointed out in this work that the successful prediction of failures translates to improved resource efficiency since there is less overhead to consider in regard to experiencing the repeated execution of tasks [6]. Elkaradawy et al. (2025) suggested the Multilayer Multi-Prediction Framework (MMPF). A two-layer approach is applied in this novel system: the first layer identifies the possible failures with the help of a voting system of multiple classifiers (DT, KNN, XGBoost), and the second layer determines the type of failure. This combination method reported an accuracy of 99.83% by compensating for the drawbacks of other models [7]. According to Eang and Lee (2025), a hybrid CNN-RNN model should be used for predictive maintenance and fault detection of industrial robot DC motor drives. CNN features are employed to access spatial features of sensor data, and RNN to access the temporal features to come up with befitting predictions of faults. The proposed model was superior to both CNN-LSTM and the traditional models in terms of accuracy and processing speed, as it demonstrates that the proposed model has an advantage in that it enables one to detect the fault at an early stage. Nevertheless, the study is limited to a smaller scope of DC motor drives, which may restrict the use to other subsystems in the industry in its current form without further assistance [8]. A hybrid scheme, named SRL-MROLS (Segmented Regressive Learning-based Multivariate Raindrop Optimized Lottery Scheduling), was suggested in the article by Sheeja Rani et al. (2025). They used the combination of Q-learning (a reinforcement learning method) and segmented regression to forecast VM failures and plan the tasks on the most trusted resources dynamically, which lowered the failure rate in the dynamic environment significantly [9]. Salanke et al. (2024) enhanced the applicability of failure prediction with a remediation strategy. They created a framework that predicts with the help of Bi-LSTM and immediately activates a dynamic task migration algorithm upon the realization of the possibility of a failure. With this method, there is continuity in service in that tasks are initially transferred to healthy nodes before a crash happens [10].

Yadav et al. (2024) tested machine failure prediction algorithms in industries, which is similar to cloud infrastructure. They compared XGBoost and LSTM and discovered that XGBoost works very well with structured tabular data, whereas LSTM works better with sequential data where failure is indicated by the temporal patterns [11]. Kumar et al. (2024) used Natural Language Processing (NLP) tools in analyzing system logs. They used the Word2Vec embedding's to transform textual log entries into a form of vectors that were fed to an LSTM network. The given approach allowed identifying failures according to the semantic meaning of log messages, but not according to the error rates only [3]. Garneedi et al. (2024) provided a comparative analysis of deep learning models that are specially implemented in cloud data centers. Their results confirmed the effectiveness of LSTM-based architectures in dealing with the high dimensionality and noise of cloud system logs, which supported the results of Gao et al. [12]. Bommala et al. (2023) took the analysis further by comparing the pattern of failures in three different traces: Google, Mustang, and Trinity. Their comparison analysis showed that the failure characteristics of various cloud environments are quite different, implying that prediction models cannot be developed as a one-size-fits-all without adaptive deep learning [13]. Asmawi et al.

(2022) conducted an extensive comparison of five classical ML algorithms (such as Random Forest and SVM) with three deep learning versions. According to their findings, the applicability of the Random Forest in the particular types of jobs was high, but Deep Learning models are more useful in the generalization of diverse failures on the task type, specifically when the issue of the class imbalance in Google Cloud Traces is present [2]. Jassas and Mahmoud (2022) were interested in analyzing job failures with the help of machine learning. Resource requests (CPU and RAM limits) were found to be good predictors of failure. They used Google Cluster Traces in their study and pointed out that tree-based models such as the Random Forests might be very accurate but less adaptable to time-series data as compared to the neural networks [14]. Saxena and Singh (2022) have created OFP-TM, an online VM failure prediction model. They have a system with an ensemble resource predictor, which predicts the health of VMs in real-time. When a failure is predicted, a so-called Selection Box mechanism isolates the risky VM and transfers its workload, paying attention to high availability [15]. Alahmad et al. (2021) developed a proactive failure-detection schedule of tasks on cloud computing grounded on the Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN). The model also forecasts termination of a task at run-time, and it also involves the use of the Integer Linear Programming (ILP) in the task of best remedy action selection. They had a maximum profitable ANN model of up to 94% accuracy in utilizing the Google cluster dataset and were in a position to reduce the wastage of resources. However, such an optimization process introduces the computational load, which can impact the scalability in large-scale cloud systems [16]. Liu et al. (2020) addressed the issue of the lack of data in the initial phases of a task. They suggested a remedy that relied on domain information mining those group jobs that had a similar pattern of resource consumption. Their model can be used to know whether a new job will be terminated by using the history of similar jobs even before generating much log data [17]. A multi-layer Bidirectional Long Short-Term Memory (Bi-LSTM) network proposed by (Gao et al., 2022) was used to predict the failure of tasks and jobs in cloud data centers. They showed that their Bi-LSTM model was suitable to learn past and future contexts of task execution to overcome the vanishing gradient issue of traditional RNNs by analyzing Google Cluster Traces. Their model recorded a prediction accuracy of about 93, which was much higher than the use of baseline methods [4].

3. Research methodology

The proposed framework is displayed in Fig. 1 and rest part of the methodology explained below.

3.1 Dataset

The Google Borg Cluster Traces 2019 dataset is used for task failure prediction in cloud computing environment in this research. The dataset is composed of workload execution logs from Google's Borg cluster management system, covering aspects of CPU usage, memory usage, scheduling behavior, task execution status, and resource consumption patterns of the workloads. This raw data includes about 405,894 task records and several attributes related to the load, including collection_id, scheduling_class, priority, assigned_memory, cpu_usage_distribution, memory_accesses_per_instruction, and scheduler information. In the dataset, there are multiple task execution events such as FAIL, FINISH, LOST, EVICT, KILL, ENABLE, and SCHEDULE, indicating the various operational states of cloud tasks for execution. The distributions of events in the data set are as follows: 92678 (FAIL), 92867 (FINISH), 59515 (LOST), 14756 (EVICT), 951 (KILL), 75907 (ENABLE), and 69104 (SCHEDULE). To make binary labels for failed and non-failed task prediction, these events were used.

3.2 Preprocessing

The preprocessing on the Google Borg Cluster Trace dataset involved:

- Cropping out irrelevant and contradictory attributes of the raw workload records.
- Handling and identification of missing values in cycles_per_instruction and memory_accesses_per_instruction by performing median imputation to keep the data set consistent.
- Transformation of task execution events to binary target classes: FAIL, LOST, EVICT, and KILL were all failure-oriented events, while FINISH, ENABLE, and SCHEDULE were all operational states.
- Categorical workload attributes are encoded into numerical representation for deep learning model compatibility.
- Applying the Standard-Scalar to normalize the numerical features so that they meet in a stable

manner and have a uniform distribution during model training.

- Preparation of the final preprocessed dataset that has engineered workload features and binary task labels for implementing ANN, LSTM, Hybrid LSTM-ANN and Hybrid GRU-ANN model.

Finally, the preprocessed Borg trace dataset was used as the input to the deep learning models for predicting task failures and assessing the performance comparison.

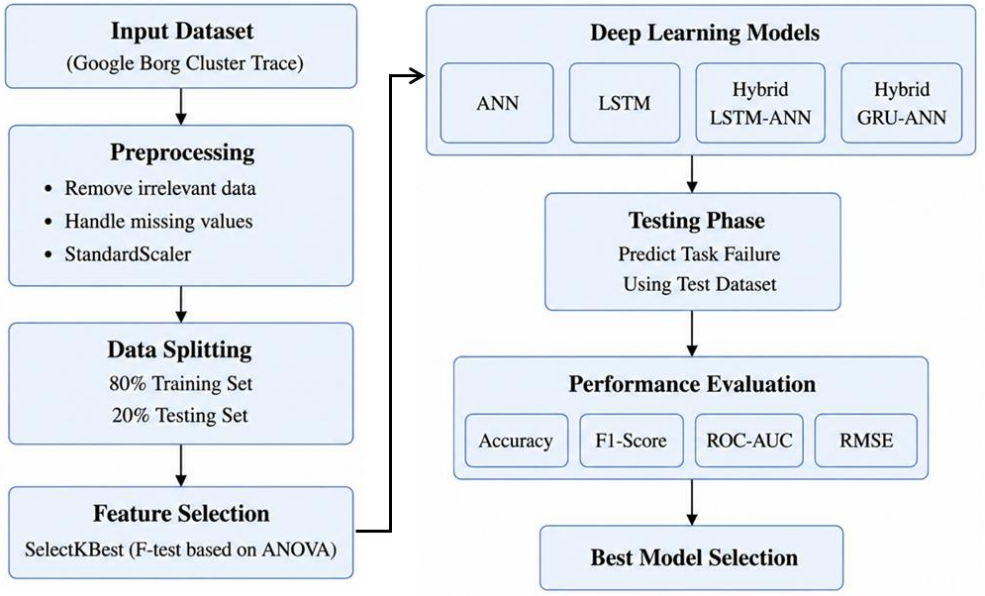

Fig. 1: Proposed Framework

3.3 Feature Selection

The feature selection was carried out to enhance the efficiency of the model, minimize feature redundancy, and maximize prediction accuracy.

Feature selection was considered using the SelectKBest method from the sklearn library. The ANOVA F-test scoring function was used to measure the statistical relationship between each input feature and the target class. Features with higher F-scores are better at distinguishing between failed and non-failed cases. The features were then evaluated based on these ranking scores; it was decided to use the top 10 most informative features evaluated and use these features for model training.

The top most features selected were collection_id, scheduling_class, collection_type, priority, alloc_collection_id, vertical_scaling, scheduler, assigned_memory, cycles_per_instruction, memory_accesses_per_instruction. The following selected features give it a more realistic workload representation and make it possible to limit the unnecessary computation overhead during training of the model. The optimized feature set was finally fed into all the deep learning models implemented to increase the prediction accuracy.

3.4 Input Representation for Sequential Models

Workload attributes are represented as structured numeric features, but the utilization pattern of resources in cloud workloads is dynamic, and a complex relationship exists among various execution parameters. These unobserved dependencies were captured by reshaping the feature vectors into a sequential array in order to be fed to the LSTM and GRU layers. These representations enable the models to acquire the complex workload status pattern related to task failure events and increase the models' ability to identify the abnormal status condition of task executions in the cloud.

3.5 Proposed Model

3.5.1 Artificial Neural Network

The Artificial Neural Network (ANN) model was designed to be used as a baseline machine learning model for cloud task failure prediction. The architecture was created with two dense hidden

layers (64 and 32 neurons with ReLU activation) and a Softmax output layer for binary classification. The ANN model was trained for 15 epochs using the Adam optimizer and categorical_crossentropy loss function with a batch size of 32. The class weights were balanced, and the training set was split into a validation set (testing set 20%) to increase the stability of the predictions and decrease over-fitting. While the ANN model had good prediction accuracy, it was found that it had difficulty in learning temporal workload dependencies found in the cloud execution data.

3.5.2 Long Short-Term Memory

To account for the temporal dependency and sequential workload behavior of the data related to task execution in a cloud environment, the Long Short-Term Memory (LSTM) model was adopted. A single LSTM layer of 64 units was followed by another dense layer of 32 neurons and ReLU activation, with an output Softmax layer for binary classification. The said model was trained with 15 epochs with an optimizer, Adam, and a loss function of categorical_crossentropy, with a batch size of 32. Class imbalance was addressed by balanced class weights. The LSTM model showed superiority in sequential learning ability and in the ability of temporal pattern analysis over the standalone ANN model.

3.5.3 Hybrid GRU-ANN Model

A hybrid of GRU and ANN was considered to enhance the prediction accuracy with minimal computation. The structure included a GRU layer with 64 units, two dense layers with 64 and 32 neurons, respectively, and a Softmax output layer with a binary classification. This hybrid model was trained for 15 epochs with a batch size of 32 using the categorical_crossentropy loss function and the Adam optimizer, with 20% of the data reserved as validation (testing). Class weights for balanced classes were used for training to enhance the reliability of predictions.

3.5.4 Hybrid LSTM-ANN Model

A hybrid LSTM-ANN model was designed, which combines the sequential learning ability of an LSTM network and the ability to learn non-linear features of an ANN layer. The architecture comprised an LSTM layer with 64 units, followed by two dense layers with 64 neurons, 32 neurons, respectively, and ReLU activation functions. For binary output task classification, a Softmax layer was applied to their output. The model was trained for 15 epochs on 32 samples (batch size) with the Adam optimizer, and the loss function used was the categorical_crossentropy. The said hybrid model has been trained for 15 epochs with the categorical_crossentropy loss function, Adam optimizer, validation_split of 0.2, and batch_size of 32.

3.6 Evaluation Metrics

To assess the effectiveness of the deep learning models implemented in this research, various classification and error-based evaluation metrics were employed to evaluate the accuracy of predictions, reliability, and generalization capacity of deep learning models for cloud task failure prediction. Evaluation metrics selected were accuracy, F1-score, ROC-AUC, Root Mean Square Error (RMSE), and confusion matrix analysis.

The accuracy can be calculated using Eq. (1) as shown below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Where, TP, TN, FP, and FN are true positive, true negative, false positive and false negative predictions respectively.

The F1-score (see Eq. 2) is used to measure the accuracy of precision and recall as there is an imbalance in the Borg trace data set between failure-oriented and non-failure task events.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2}$$

The Root Mean Square Error (RMSE) is used to measure prediction error and model's reliability. It is considered: the lower the RMSE, the more accurate the prediction and the less the classification error.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

Where, y_i is the actual value and $\hat{y}_i$ is predicted value.

The classification ability of the models to discriminate between failed and non-failed task was evaluated by using the ROC-AUC (Receiver Operating Characteristic – Area Under-Curve) metric.

Here, the higher ROC-AUC ensures better discrimination. The ROC-AUC determines how fine a model separates positive and negative classes across all possible classification thresholds.

The Confusion Matrix was also used to visualize the classification performance of the models by comparing correctly and incorrectly predicted failed and non-failed tasks.

4. Results and Discussion

The Google Borg Cluster Traces 2019 dataset has been utilized to evaluate and compare the performance of the proposed deep learning models for predicting cloud task failure. The implemented architectures are ANN, LSTM, Hybrid LSTM-ANN and Hybrid GRU-ANN. Experiments were simulated on the Python environment with TensorFlow.

4.1 ANN

The model converged towards stable values during learning, and exhibited strong classification capabilities for the prediction of cloud loads. The ANN model had an accuracy of 92.95%, F1-score of 0.8523, ROC-AUC of 0.97 and RMSE of 0.2408 (see Fig. 10). The confusion matrix analysis (see Fig. 3) revealed that the ANN model was able to classify most failed and non-failed tasks. Although there were misclassifications, this is happened due to a limitation of the model to capture and represent temporal workload dependencies. The prediction performance of the ANN model was satisfactory, but its learning capability was relatively low than the hybrid deep learning architectures.

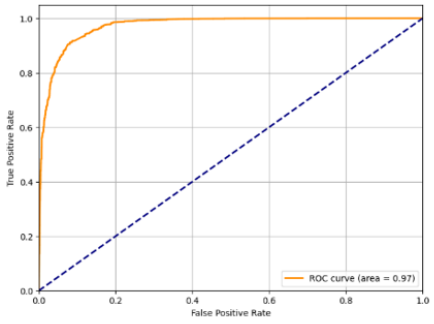

Fig. 2: ROC curve of Artificial Neural Network (ANN)

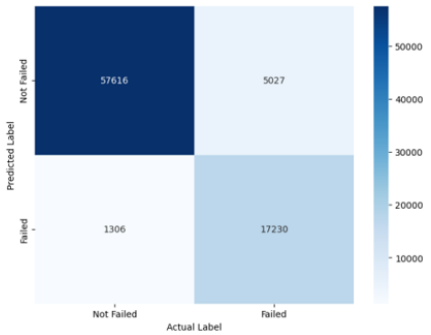

Fig. 3: Confusion Matrix of ANN

4.2 LSTM

The LSTM model is a memory-based model, the model exhibited better capability of temporal learning than the individual ANN model. The LSTM architecture attained accuracy level of 91.63%, an F1-Score of 0.8370, ROC-AUC value of 0.98 and RMSE value of 0.2499 (see Fig. 10). The confusion matrix (see Fig. 5) suggests that the behavior of the workload sequence was well learned by the model and the classification of events of fault was enhanced.

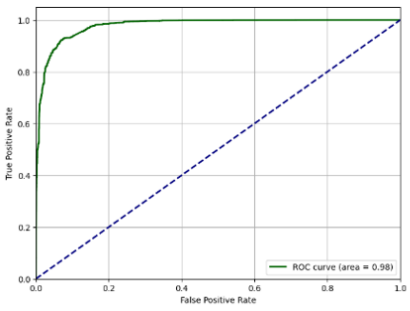

Fig. 4: ROC curve of LSTM

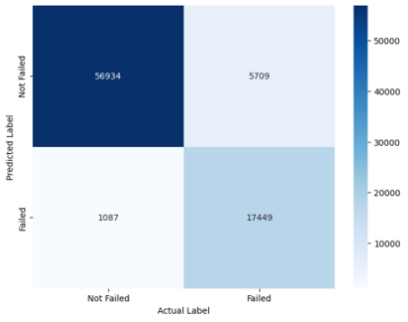

Fig. 5: Confusion Matrix of LSTM

4.3 Hybrid GRU-ANN Model

The Hybrid GRU-ANN model was implemented for efficiency of prediction and yet keeping the

classification strong. The model's architecture was based on two deep learning models: the sequential learning capacity of the GRU model and the nonlinear feature learning capacity of the ANN layers. The Hybrid GRU-ANN model achieved an accuracy of 95.36%, an F1-score of 0.9028, ROC-AUC value of 0.9906, and an RMSE value of 0.1865 (see Fig. 10). Confusion matrix analysis (see Fig. 7) demonstrated that the model was good at classification for cloud workload events and maintained stability in the convergence, which also had lower predicting errors.

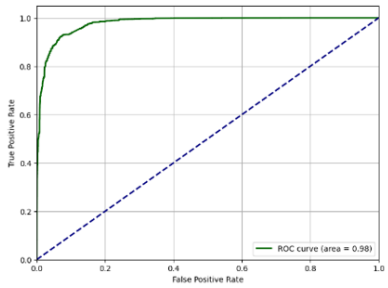

Fig. 6: ROC curve for Hybrid GRU-ANN

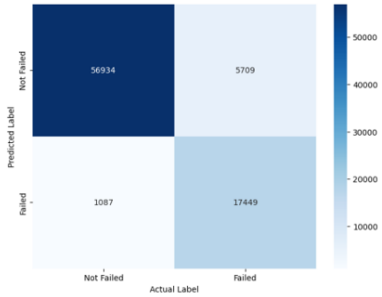

Fig. 7: Confusion Matrix of Hybrid GRU-ANN

Despite weaker performance when compared with the Hybrid LSTM-ANN model, the GRU based hybrid architecture proved to be of lower complexity and quicker convergence ability without compromising the prediction consistency compared to rest models.

4.4 Hybrid LSTM-ANN Model

The proposed LSTM-ANN model achieved an accuracy of 96.08%, F1-score of 0.9173, ROC-AUC value of 0.9925, and the lowest RMSE value of 0.1721 (see Fig. 10). The confusion matrix analysis (see Fig. 9) showed very accurate classification in terms of highly low misclassification rates both for failed and non-failed tasks.

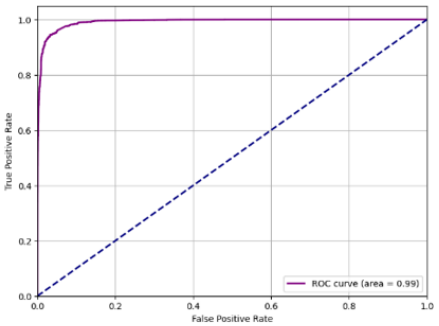

Fig. 8: ROC Curve for Hybrid LSTM-ANN

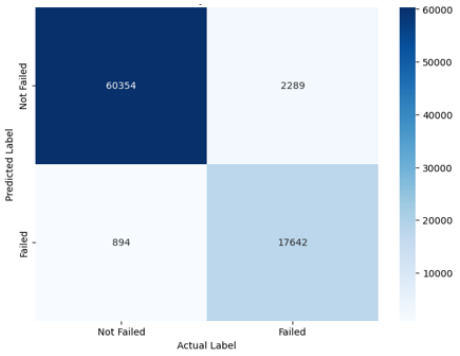

Fig. 9: Confusion Matrix for Hybrid LSTM-ANN

Further, the model exhibited high generalization ability as seen in the limited over-fitting of the training and validation performance. The experimental results demonstrated that the sequential learning and dense NN fusion significantly enhances the cloud task failure prediction accuracy performance.

4.5 Comparative Performance Analysis

The comparison of the results obtained by all models is displayed in Fig. 10. The Hybrid LSTM-ANN model showed the best performance across all implemented models, and it achieved the best high values of Accuracy, F1-Score and ROC-AUC and the lowest RMSE value. Another model in the current study, the Hybrid GRU-ANN model, also showed very competitive performance with lower complexity and stay on second position. The success of hybrid architectures stems primarily from their capability to capture both temporal workload dependencies as well as non-linear relationships between features by combining information from cloud execution data. Compared with the standalone ANN model which did not have the sequential learning capability, the standalone LSTM model showed less nonlinear feature abstraction than the hybrid architecture models.

Moreover, the ROC-AUC results of all the setups of the models were higher than 0.97 which shows good discriminating capability of each model in discriminating failed and non-failed cloud tasks. The better for the hybrid architectures in terms of lower RMSE values also indicated higher prediction reliability and lower classification error. For clear understanding, the results are also given in Table 1.

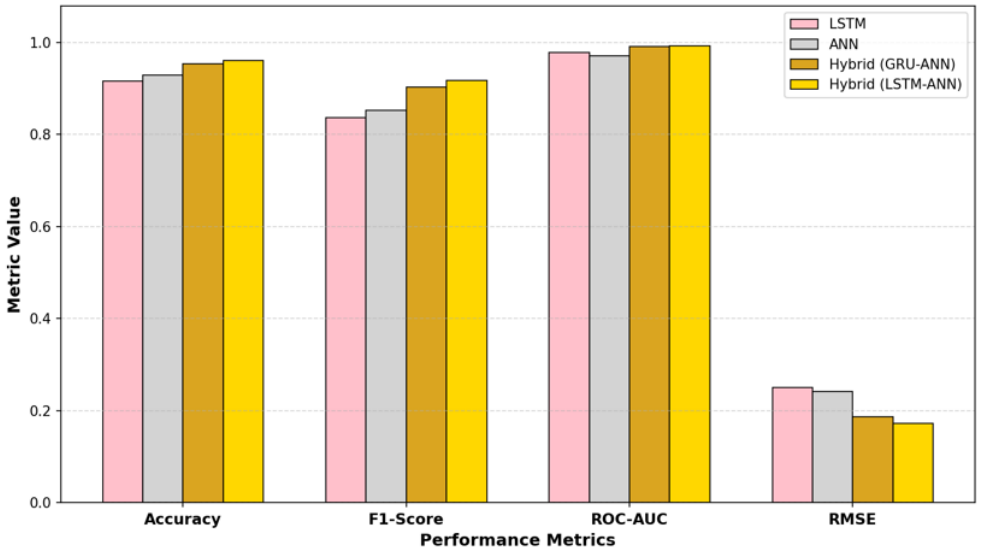

Fig. 10: Performance Comparison of Deep Learning models

Table 1: Comparative Performance Analysis of Models

Model	Accuracy	F1-Score	ROC-AUC	RMSE
ANN	0.9295	0.8523	0.9716	0.2408
LSTM	0.9163	0.8370	0.9778	0.2499
Proposed Hybrid LSTM-ANN	0.9608	0.9173	0.9925	0.1721
Hybrid GRU-ANN	0.9536	0.9028	0.9906	0.1865

Table 2: Comparison of Proposed Model with Others

Study	Dataset	Model	Accuracy (%)
Gao et al. (2022)	Google Cluster Traces 2019	Bi-LSTM	93.0
Alahmad et al. (2021)		ANN-CNN	94.0
Proposed Model		Proposed Hybrid LSTM+ANN	96.08

In Table 2, the comparison of the proposed Hybrid LSTM-ANN with other existing models is also displayed. The comparison with other studies proves that the proposed framework outperforms in terms of prediction accuracy for cloud task failure prediction. These findings validate the sequential learning-approach along with combined non-linear feature extractions approach in predicting workload incapacity in cloud computing environment. The hybrid LSTM-ANN model effectively integrated the two layers of the model, namely LSTM and ANN to learn the temporal dependency and acquire the nonlinear classification ability respectively in order to well capture the complex workload patterns and abnormal execution behaviors within cloud environments.

In the end, the experimental results confirm that hybrid Deep Learning LSTM-ANN architecture is an excellent and reliable approach for proactively predicting task failures in cloud computing.

5. Conclusion and Future Scope

In this paper, we have proposed a deep learning hybrid model for task failure prediction in cloud computing using Google Borg Cluster Traces 2019 dataset. This study also conducted a comparative performance analysis of several deep learning techniques for cloud task failure prediction using the Google Borg Cluster Trace dataset. The proposed framework analyzed cloud workload behavior using artificial neural networks (ANN) and long short-term memory (LSTM). To improve prediction

reliability and overall system performance, the study incorporated preprocessing, feature selection, handling of dataset imbalance, and deep learning-based predictive analysis. The experimental results confirmed that proposed hybrid deep learning framework offer meaningful improvements in cloud task failure prediction compared to standalone models. The hybrid LSTM-ANN model achieved the best overall performance, with an accuracy of 96.08%, an F1-score of 0.9173, a ROC-AUC value of 0.9925, and the lowest RMSE of 0.1721. The hybrid GRU-ANN model also delivered competitive results with lower computational complexity and stable convergence and stay on second position. These findings demonstrate that combining sequential learning with nonlinear feature extraction is an effective approach for detecting abnormal workload behavior in cloud environments in case of LSTM-ANN. These strengths can support failover management, resource optimization, and improved service reliability for cloud providers operating at large scale.

Future research could incorporate advanced optimization techniques to enable automatic feature selection and hyper-parameter tuning, with the goal of further improving prediction accuracy and reducing model complexity. Additional deep learning architectures, including attention-based networks, Transformer models, and ensemble hybrid designs, could also be explored to better capture complex workload execution patterns and long-term temporal dependencies in cloud systems.

References

1. T. Hagras and G. A. El-Sayed, “A fault-tolerant and load-balancing scheduler for independent tasks on cloud-based virtual machines,” Clust. Comput., vol. 29, no. 1, p. 61, Feb. 2026, doi: 10.1007/s10586-025-05857-1.
2. T. N. Tengku Asmawi, A. Ismail, and J. Shen, “Cloud failure prediction based on traditional machine learning and deep learning,” J. Cloud Comput., vol. 11, no. 1, p. 47, Sep. 2022, doi: 10.1186/s13677-022-00327-0.
3. T. Kumar, Yashika, A. Singhal, Yashvardhan, and R. Priyadarshini, “Early System Failure Detection through System Log Analysis: An LSTM Approach,” in 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India: IEEE, Jun. 2024, pp. 1–7. doi: 10.1109/ICCCNT61001.2024.10725393.
4. J. Gao, H. Wang, and H. Shen, “Task Failure Prediction in Cloud Data Centers Using Deep Learning,” IEEE Trans. Serv. Comput., vol. 15, no. 3, pp. 1411–1422, May 2022, doi: 10.1109/TSC.2020.2993728.
5. S. Aldomi, H. Suleiman, A. Shatnawi, and L. Alawneh, “A deep learning approach for early prediction of task failures in cloud computing environments,” Syst. Soft Comput., vol. 8, p. 200442, Jun. 2026, doi: 10.1016/j.sasc.2026.200442.
6. W. Wu, “Fault Detection and Prediction in Models: Optimizing Resource Usage in Cloud Infrastructure,” Feb. 20, 2025, In Review. doi: 10.21203/rs.3.rs-6059985/v1.
7. A. Elkaradawy, A. Elshenawy, and H. Harb, “Enhancing Cloud Job Failure Prediction With a Novel Multilayer Voting-Based Framework,” IEEE Access, vol. 13, pp. 140600–140613, 2025, doi: 10.1109/ACCESS.2025.3593808.
8. C. Eang and S. Lee, “Predictive Maintenance and Fault Detection for Motor Drive Control Systems in Industrial Robots Using CNN-RNN-Based Observers,” Sensors, vol. 25, no. 1, p. 25, Dec. 2024, doi: 10.3390/s25010025.
9. S. S. Rani, O. Alfawaz, and A. M. Khedr, “A robust fault-tolerant framework for VM failure predication and efficient task scheduling in dynamic cloud environments,” J. Netw. Comput. Appl., vol. 244, p. 104340, Dec. 2025, doi: 10.1016/j.jnca.2025.104340.
10. V. S. Salanke, V. N. Kowshik, H. P. G, M. V, and P. H, “Task Failure Prediction and Migration in Cloud Environment,” in 2024 1st International Conference on Communications and Computer Science (InCCCS), Bangalore, India: IEEE, May 2024, pp. 1–6. doi: 10.1109/InCCCS60947.2024.10593580.
11. D. K. Yadav, A. Kaushik, and N. Yadav, “Predicting machine failures using machine learning and deep learning algorithms,” Sustain. Manuf. Serv. Econ., vol. 3, p. 100029, 2024, doi: 10.1016/j.smse.2024.100029.
12. P. Garneedi, R. K. Kollipara, G. R. Kurri, and M. Mane, “Task Failure Prediction in Cloud Data Centers Using Deep Learning,” vol. 13, no. 3, 2024.
13. H. Bommala, U. M. V., R. Aluvalu, and S. Mudrakola, “Machine learning job failure analysis and prediction model for the cloud environment,” High-Confid. Comput., vol. 3, no. 4, p. 100165, Dec. 2023, doi: 10.1016/j.hcc.2023.100165.
14. A. K. Mohammed et al., “Predictive Failure Detection in Cloud Infrastructure Using Multivariate Telemetry Log Analysis with Temporal Convolution and Attention-Based Deep Learning,” in 2025 3rd International Conference on Cyber Resilience (ICCR), Dubai, United Arab Emirates: IEEE, Jul. 2025, pp. 1–8. doi: 10.1109/ICCR67387.2025.11292090.
15. D. Saxena and A. K. Singh, “OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments,” J. Supercomput., vol. 78, no. 6, pp. 8003–8024, Apr. 2022, doi: 10.1007/s11227-021-04235-z.
16. Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, and A. S. A.-M. Al-Ghamdi, “Technical Study of Deep Learning in Cloud Computing for Accurate Workload Prediction,” Electronics, vol. 12, no. 3, p. 650, Jan. 2023, doi: 10.3390/electronics12030650.
17. C. Liu, L. Dai, Y. Lai, G. Lai, and W. Mao, “Failure prediction of tasks in the cloud at an earlier stage: a

solution based on domain information mining," Computing, vol. 102, no. 9, pp. 2001–2023, Sep. 2020, doi: 10.1007/s00607-020-00800-1.