Predict Task Failure In VM Clouds With Alibaba Cluster DataSet Based On GRU

Ameer Umer
Dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
ameerumer1947@gmail.com

Hussain Baksh Khan Joiya
Dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
hussainjoiya714@gmail.com

Muhammad Atif
Dept. of Data Science
Islamia University of Bahawalpur
Bahawalpur, Pakistan
atifahmadani68@gmail.com

Sana Tariq
Dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
Sana.tariq@iub.edu.pk

Aoun Muhammad
Dept. of Information and Communication Engineering
Islamia University of Bahawalpur
Bahawalpur, Pakistan
aoun.muhammad@iub.edu.pk

***Abstract*—Non-productive tasks failures, unexpected delays in cloud environments may cause inefficient use of resources, and therefore less reliability for services. Catastrophic failures can be avoided by accurately predicting the outcome of each task performed by cloud resources. In this paper a comparative study is proposed on the application of deep learning models to predict the status of a cloud task based on Alibaba Cluster Dataset is proposed in this paper. The subsets used from the data set were used for preprocessing and feature selection, resulting in a balanced data set with 112,928 task instances and four task status classes for training and evaluation of the model. The study compares the performance of recurrent models, hybrid models, and transformer-based models such as LSTM, ANN, GRU, GRU-LSTM, GRU-ANN, Temporal Fusion Transformer (TFT), Informer, and Tab Transformer. The results of the experiments revealed that the GRU model achieved the best precision of 97.3%, while the LSTM models and the ANN models showed similar performance. The models trained on a hybrid were also the competitive ones; however, these were not produced in an improvement of the standalone architecture of GRU. Transformer architectures were compared with recurrent architectures for selected workload features and performed competitively, but did not achieve the highest accuracy among the various models tested, the Tab Transformer being the lowest of the lot. The results show that GRU successfully achieves an acceptable predictive accuracy and model complexity for Alibaba workload traces for the status prediction task on the cloud computing platform.**

***Index Terms*—Cloud Computing, Task Status Prediction, Task Failure Prediction, Deep Learning, Gated Recurrent Unit (GRU), Transformer Models, Alibaba Cluster Dataset**

I. Introduction

A. The Cloud Failure Problem

From real-time analytics platforms to enterprise-critical applications, cloud services are an essential part of today's digital world. The thrust of these applications is to run large quantities of jobs on virtual machines or cloud resources. However, failures can occur due to resource source contention , memory overload, scheduling issues, or hardware failures, and these failures can negatively impact system performance and service reliability [1]. This can result in lost compute resources, dissatisfaction with service level agreements (SLAs), and a lower service quality level.Due to the magnitude of cloud operations, task failures in large-scale cloud may be a relatively small number, but can translate into a large number of task failures on a daily basis. Knowing what might go wrong helps cloud management systems to act in the correct order when needed to migrate a workload, allocate suitable resources to it, or implement a fault-tolerance strategy before service interruption. In the context of critical applications like healthcare data processing, finance, or any other field demanding steady service, predicting tasks becomes a critically sensitive function.Therefore, the area of task prediction has become crucial in critical applications such as healthcare data processing, financial services, and other fields where continuous availability is required [2].

B. Limitations of Existing Approaches

It is difficult to predict cloud task failure using a classification problem because it relies on the metadata which is generated during the execution of the cloud tasks at runtime in a cloud environment. The terminal state of a task can depend on the change over time of various resource loads: CPU, memory, I/O activity [12]. Most of the approaches used in traditional machine learning do not consider temporal relationships in the workload traces, as they consider each observation independent [14]. To deal with this, recurrent deep learning models use recurrence with memory to let existing past information carry over to inform the present. These architectures have been found to perform well in capturing

DOI: https://doi.org/10.67535/tsp.000003.026
Published under the ICACNC 2026 Proceedings

patterns from sequential data and mitigating vanishing gradients, particularly GRU and LSTM [8]. The multi-layer hybrid architecture allows for the representation of richer interaction among more complex features but is not necessarily more predictive in all cases for all data sets. The effectiveness of a model is not only based on the model itself but is affected by the distribution of status classes for tasks, as well as the workload trace and its properties.

C. Our Contribution

We systematically study various deep learning models for predicting the status of a cloud VM in this work, based on the Alibaba Cluster Dataset. A "balanced" subset of the data set is prepared in order to perform a fair evaluation of all classes of tasks. Our one-dimensional models are trained and evaluated using the same experimental conditions: recurrent, hybrid, transformer, such as LSTM, ANN, GRU, GRU-LSTM, GRU-ANN, TFT, Informer, and Tab Transformer. Comparative analysis reveals that the GRU model outperforms hybrid and transformer-based models with a record 97.3% accuracy. In addition, we compared a detailed analysis of every model according to various performance measures and compared their performance in the prediction of the cloud task status.

D. Problem Formulation

The problem of predicting the status of a cloud task can be defined as a multi-class classification problem [2]. A feature vector based on attributes of resource usage during the execution of the task is built for each task instance, and a classification to one of a set of four terminal states is sought: Failed, Interrupted, Running, and Terminated. The Alibaba Cluster Dataset contains information at the task level that describes how much of a resource is used by a task and its execution characteristics, which can be utilized to learn patterns related to various task outcomes [12]. This work aims at assessing the effectiveness of the recurrent, hybrid and transformer-based deep learning architectures against these workload features in predicting task status.

II. Related Work

A. Machine Learning for Task Failure Prediction

As the major interest in cloud task failure prediction lies in the potential to enhance resource utilization and service dependability, machine learning has sparked much interest. Previous work tended to place value on using classic machine learning methods and considering only the static representations of features. Gollapalli et al. [3]tested a variety of machine learning techniques on traces from Google's clusters and found that ensemble models were highly predictive and were still sensitive when using feature engineering techniques. In their work, Tengku Asmawi et al. [1] contrasted traditional machine learning and deep learning approaches to cloud failure prediction, and discussed the benefits of adopting deep learning models for learning complex workload patterns. Saxena et al. [12]. conducted a detailed survey of techniques used for making workload prediction and found that there is an improvement in predictive ability with models incorporating temporal information but this improvement can come at the cost of having different performance levels under various workload distributions. Considering the tight coupling of resource usage and failure prediction, Bal et al. [6]came up with a combined framework that combines resource allocation, task scheduling and making failure prediction based decisions for cloud environments.

B. Deep Learning Architectures

In cloud computing environments, recurrent neural networks have proven to be suitable for modeling workload patterns, because they are capable to learn dependency of the data that is generated sequentially. LSTM networks solve the vanishing gradient problem in traditional RNNs by introducing gating functions that control the flow of information between the time steps of a network [10]. GRU's predictive power on a variety of sequence-learning tasks is competitive, and it is much simpler and has fewer parameters. Ali et al. [8] showed that deep learning models based on GRU can be quite effective in predicting cloud workloads, especially considering that these models can capture the complex workload behaviour. Jeon et al. [9] discussed hybrid deep learning approaches to container scheduling and they claimed that, in some cases, deeper architectures can lead to a better representation of workload characteristics. Likewise, Sham and Vidyarthi [11] presented a hybrid machine learning approach for admission control in fog-integrated cloud environments, showing the benefits of using multiple machine learning techniques for cloud management applications.

C. Hybrid and Ensemble Approaches

Much research has focused on hybrid and group methods to enhance the reliability and robustness of cloud computing systems. A similar hybrid system to improve the reliability and fault-tolerance of cloud systems was explored by Shahid et al [4]. Elkaradwy et al. [15] proposed a multi-layer voting system for predicting the failure of jobs in the cloud and showed the enhanced generalization performance when applied to different workloads. Simaiya et al. [7] addressed cloud load balancing using optimization techniques and deep learning models, while Lilhore et al. [16] introduced a hybrid deep learning cloud-edge resource optimization framework. Khan et al. [5] and Zhou [17] proposed a hybrid machine learning algorithm to manage cloud resources management and task scheduling. These works demonstrate the increased interest in using a hybrid approach to solve complex cloud computing problems, but present the issue of how effective a hybrid architecture will be as a function of the workload and application domain.

III. Methodology

The overall procedure used in this study is shown in Fig. 1. First, the data is collected from Alibaba Cluster Dataset, then it preprocessed, feature prepared, model trained and performance checked. All models are trained using the same experimental setup, and evaluated under the same experimental settings to

ensure a fair comparison. The workflow offers a complete model for forecasting the status of a cloud task using deep learning architectures that include both recurrent, hybrid, and transformer models.

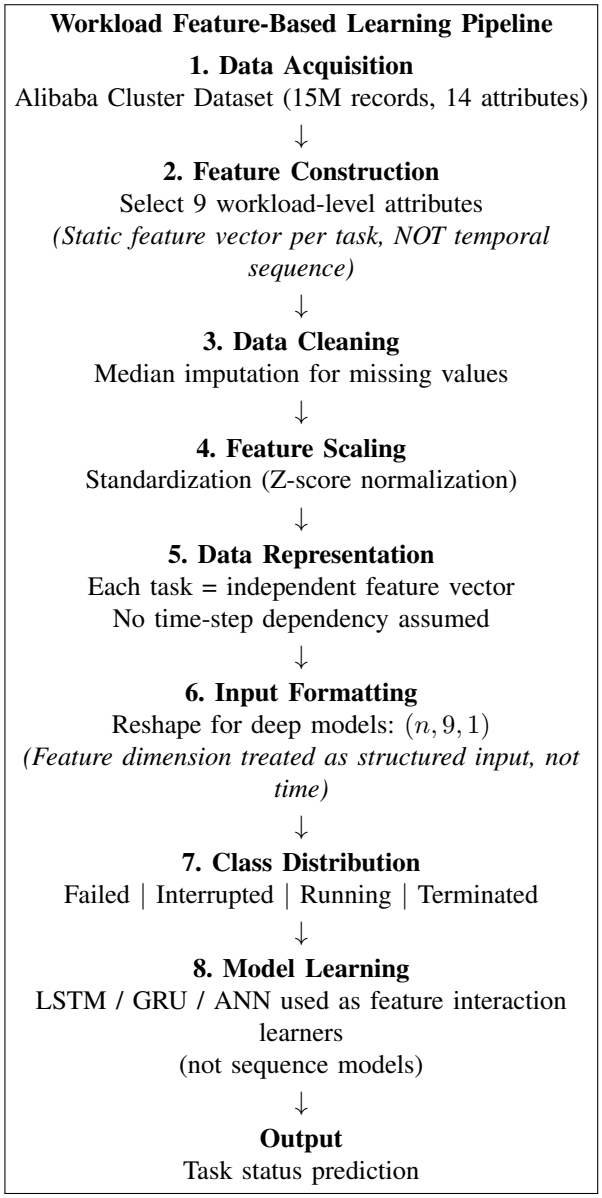

Fig. 1. Workload feature-based learning pipeline for task status prediction.

A. Dataset

The data set is very imbalanced with the task status classes. A balanced subset was built by randomly sampling 28,232 instances per task status set for fair evaluation across all task status categories. This led to a well-balanced dataset where the Failed, Interrupted, Running and Terminated classes had equal numbers of task instances; namely 112,928. The balanced dataset produced is 112,928 instances of tasks, with 28,232 instances per status class of tasks. A large-scale cloud workload trace, Alibaba Cluster Dataset (ACD) (`batch_instance.csv`), which is publicly available, was used to perform experiments. The original data is a dataset of about 15 million task instance records and 14 attributes. Table **??**summarizes the data set configuration that was used in this study and shows that four columns were dropped as identifiers and nine numerical workload-related features were retained: `task_type`, `start_time`, `end_time`, `seq_no`, `total_seq_no`, `cpu_avg`, `cpu_max`, `mem_avg`, and `mem_max`. Table I summarizes the dataset configuration used in this study.

TABLE I
ALIBABA CLUSTER DATASET CONFIGURATION

Property	Details
Source	Alibaba Cluster (`batch_instance.csv`)
Total Records	15,000,000
Balanced Sample	112,928 (28,232 per class)
Numeric Features	9
Target Classes	4
Class Labels	Failed, Interrupted, Running, Terminated
Train / Val / Test	80% / 10% / 10%
Training Set Size	90,342
Validation Set Size	11,293
Test Set Size	11,293

B. Preprocessing and Feature Preparation

Median imputation was used for data imputation. The nine numerical features were normalized with Z score normalization to give equal weighting of each feature during training of the models. Standardized Feature Value is calculated as:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

Here, x denotes the score of the original feature, and μ and σ denote the mean and standard deviation of the feature.

The tasks were compiled and represented individually as workload feature vectors for each workload instance (9 attributes). The input data were pre-processed into the shape (n, 9, 1) to make them compatible with architectures of recurrent neural networks. Not all of these nine features are observations obtained at consecutive time steps: These are features of the task workload. Thus, true temporal sequences were not modeled by the recurrent models that were evaluated in this study, but rather complex feature interactions among workload attributes for task status prediction were learned.

C. Model Architectures

To achieve a fair comparison, all models were trained in the same way. All architectures were trained using the same parameters: Adam optimizer, sparse categorical cross-entropy loss, batch size of 32, and early stopping (patience $= 3$). The maximum number of epochs used was n=20 for each of the models with all other training parameters fallowing the same.

LSTM recurrent baseline model was employed in the form LSTM. The LSTM architecture was used in this input which does not actually describe a temporal sequence but to linearize complex relationships among attributes of the workload in the LSTM's gated memory.The gated memory helps in retaining

and selectively modifying memory during learning, and the LSTM architecture was used to allow learning of complex relationships among the attributes of the workload, though the input was not actually in a temporal sequence.

ANN feedforward baseline was for ANN. It comprises all layers which are fully connected to each other and have RELU activation functions, and is trained to learn direct links between workload features without any recurrent processing.

GRU uses update and reset gates to control the passage of information, but advocates the fact that it's easier to work with than LSTM. GRU is efficient, as it has only few parameters to be trained and is less complex than CNN and LSTM, in terms of computation.GRU has less number of trainable parameters and lower computational complexity which makes it efficient for learning feature interactions from the data corresponding to workload. It can effectively learn the features of the tasks and minimize training time, which makes it well suited for predicting status of a task in cloud.The mathematical formulation of the GRU is given as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1},\, x_t] + b_z) \tag{2}$$

$$r_t = \sigma(W_r \cdot [h_{t-1},\, x_t] + b_r) \tag{3}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1},\, x_t] + b) \tag{4}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{5}$$

GRU-LSTM is a hybrid model: a model consisting of GRU layer followed by an LSTM layer. The model is designed to achieve complementary feature learning through a single architecture combining the learning power of the two recurrent architectures.

GRU-ANN consists of a GRU-based feature Extraction stage followed by fully connected neural network layers for classification. In this architecture, the GRU extracts high–level feature representations from the workload attributes and uses dense layers to make the task status prediction.

TFT is a transformer-based architecture whose task is to capture the complex interactions between features with attention. In this study, TFT was examined to see if the approach of attention-based learning can enhance the task status prediction from workload features.

Informer is an efficient transformer architecture equipped with probabilistic sparse self-attention to achieve efficient learning, while retaining the capability to learn. It was added to evaluate the performance of lightweight transformer models in predicting the states of tasks in the cloud.

Tab Transformer is designed to be applied to tabular data through the transformer-based attention mechanism. The model was tested to check its capability to capture dependencies between attributes related to the workload and to increase the classification accuracy.

IV. Experimental Setup

In the following, all experiments were executed in Python 3.12 implementing the deep-learning TensorFlow 2.20 + Keras and the preprocessing + evaluation Scikit-learn 1.3. Model training was conducted with the help of one NVIDIA T4 GPU with Google Colab. This is a balanced dataset and was divided into a training-validation-test set ratio of 80/10/10, to produce 90,342 training instances, 11,293 validation instances and 11,293 test instances. Three measures used to evaluate performance were precision, recall, accuracy and weighted-average F1-score which are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$F_1 = \frac{2PR}{P + R} \tag{9}$$

V. Results and Discussion

A. Overall Performance

Table II presents the performance of all models on the 112928 instance balanced test set. All metrics are real experimental results from model training on the Alibaba Cluster Dataset.

B. GRU Model Analysis

For further examination and analysis of the optimal performance of the optimal model chosen, confusion matrix analysis, training curve and ROC curve were conducted. The visualizations offer a glimpse of the classification behavior, learning process and discriminative ability of the GRU model.

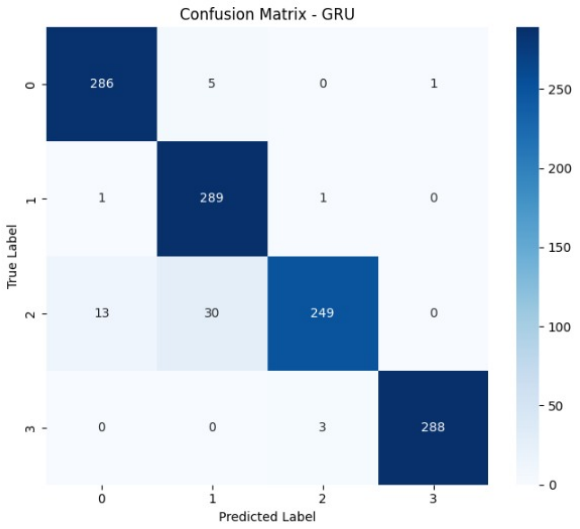

Fig. 2. Confusion matrix of the GRU model

Fig. 2 shows the confusion matrix for GRU model. Observing the results, it can be seen that most of the task instances are well classified for all the four classes of task status. A small sample of misclassified samples is an indication of the good learning of discriminative patterns by the model from

TABLE II
PERFORMANCE COMPARISON OF ALL MODELS

Model	Accuracy	Precision	Recall	F1_score	AUC
LSTM	0.97±0.013	0.980±0.01	0.97±0.013	0.97±0.010	0.996±0.002
ANN	0.97±0.01	0.982±0.001	0.972±0.01	0.975±0.004	0.997±0.000
GRU	0.973±0.01	0.980±0.01	0.97±0.01	0.975±0.01	0.997±0.000
GRU-ANN	0.95±0.02	0.98±0.01	0.95±0.02	0.96±0.02	0.997±0.001
GRU-LSTM	0.96±0.01	0.98±0.004	0.96±0.01	0.97±0.005	0.997±0.001
TFT	0.92±0.21	0.97±0.01	0.92±0.021	0.93±0.02	0.995±0.002
Informer	0.92±0.02	0.97±0.004	0.92±0.02	0.94±0.01	0.99±0.0001
Tab Transformer	0.37±0.198	0.18±0.098	0.37±0.198	0.24±0.13	0.59±0.05

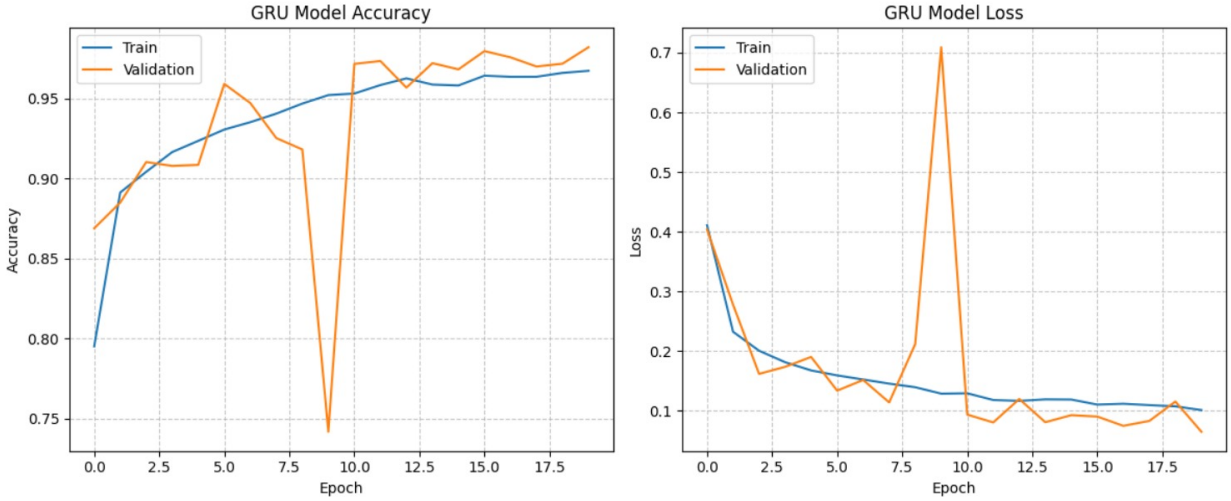

Fig. 3. Accuracy and loss curves of the GRU model

TABLE III
COMPUTATIONAL COST COMPARISON

Model	Train Time	Latency	Params
LSTM	420.00±12.284	0.0001±0.0	30804
ANN	148.40±1.531	0.0001±0.0	9796
GRU	409.8±13.223	0.0001±0.0	23454
GRU-ANN	350.98±4.25	0.0001±0.0	17284
GRU-LSTM	398.61±1.33	0.0001±0.0	46148
TFT	198.33±2.51	0.0001±0.0001	144356
Informer	184.06±2.23	0.0001±0.0	21316
Tab Transformer	271.19±7.39	0.0002±0.0	87812

the workload features and balanced performance of the model across the task statuses.

In the figure,3 the training and validation accuracy curve and loss curve for GRU model are plotted respectively. Loss decreases during training, and both the accuracy curves rise almost as time goes by and get closer, and the accuracy curve at convergence, while both curves are steady. The high similarity in training and validation accuracy indicates low level of overfitting.

Fig. 4 shows the ROC curve of the GRU model. The curve is near the upper left corner, showing good discriminative capability between the different task status classes. As reported in Table II the high AUC value also indicates the effectiveness

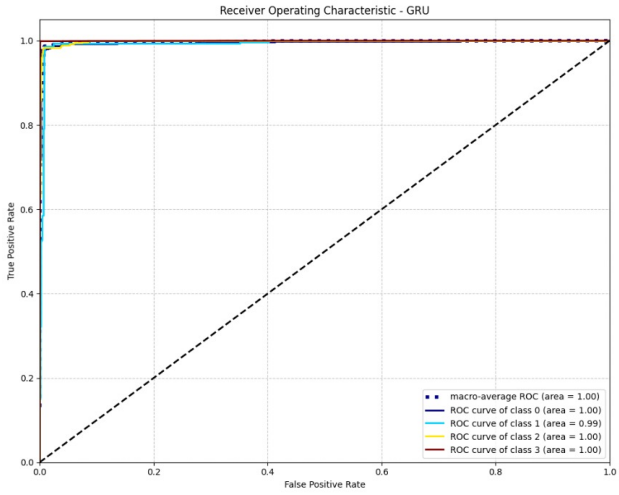

Fig. 4. ROC curve for the GRU Model

of the GRU model for the prediction of cloud task status, providing further proof of the success of the model.

C. Discussion

As seen from the results, all models were able to learn useful patterns from the Alibaba Cluster Dataset, however the per-

formance might vary when feeding different architectures of the dataset. Overall, the GRU model had the highest accuracy among all models tested at 97.3%, and good precision, recall, and F1-scores. However, LSTM also had a good performance with 97% accuracy and ANN had a similar accuracy. The small variation between these models indicates that when it comes to reliable prediction of task status enough information is contained in the selected workload features.

There was no significant improvement over the standalone recurrent models in the hybrid models. Compared to the GRU-ANN model, GRU-LSTM improved the accuracy to 96%. Results obtained by both models turned out to be satisfactory, but the extra layers did not increase the accuracy of classification. The model of grouping the data in a balanced manner remained the best option for the use of a standalone GRU for the data set used in the current study.

It is clear that the Transformer based models achieved lower results than the recurrent models. The accuracy of TFT and Informer was about 92%, while Tab Transformer was the least accurate with an accuracy of 37%. These results indicate that for the problem of workload prediction, and for the respective features and dataset setup used in this study, recurrent-based architectures over-performed the transformer-based ones.

VI. Conclusion and Future Work

In this work, the features were extracted from a training test set using deep learning models to predict a task's status in the Alibaba Cluster Dataset. For a fair comparison between the architectures, 112,928 task instances were used, which is a balanced dataset. Results obtained indicate that the best accuracy of 97.3% was obtained for the GRU model followed by the LSTM and ANN with almost the same scores. While the results of the hybrid models were competitive, the models GRU-LSTM and GRU-ANN were not able to outperform the GRU alone. The transformer-based models (namely TFT and Informer) were less accurate, and Tab Transformer had the worst performance out of all the considered models. Based on these findings, GRU achieved the highest accuracy in the experimental setup among the three operating on the selected workload data as it provides good predictions with less complexity due to an absence of a hybrid architecture.

The results are encouraging, but a number of areas have yet to be explored. Future experiments will be performed on a larger set of the Alibaba Cluster trace, plus other cloud workload datasets, to see if the performance of the models as observed are still consistent across different environments. The effect of bidirectional recurrent networks and attention-based mechanisms can also be evaluated to gain insight into the possibility of further predictive gains through these mechanisms. The adoption of adaptive / online learning methods, which can adapt to varying workload behaviour over time, is also of interest. Model interpretation techniques also can be integrated to gain understanding in prediction decision making and assists better use of these models in the cloud resource management systems.

References

[1] T. N. Tengku Asmawi, A. Ismail, and J. Shen, "Cloud failure prediction based on traditional machine learning and deep learning," *J. Cloud Computing*, vol. 11, no. 1, p. 47, 2022.

[2] S. Aldomi, H. Suleiman, A. Shatnawi, and L. Alawneh, "A deep learning approach for early prediction of task failures in cloud computing environments," *Systems and Soft Computing*, p. 200442, 2026.

[3] M. Gollapalli *et al.*, "Task failure prediction using machine learning techniques in the Google cluster trace cloud computing environment," *Mathematical Modelling of Engineering Problems*, vol. 9, no. 2, 2022.

[4] M. A. Shahid, M. M. Alam, and M. M. Su'ud, "Achieving reliability in cloud computing by a novel hybrid approach," *Sensors*, vol. 23, no. 4, p. 1965, 2023.

[5] A. Khan *et al.*, "EcoTaskSched: a hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments," *Scientific Reports*, vol. 15, no. 1, p. 12296, 2025.

[6] P. K. Bal *et al.*, "A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques," *Sensors*, vol. 22, no. 3, p. 1242, 2022.

[7] S. Simaiya *et al.*, "A hybrid cloud load balancing and host utilization prediction method using deep learning and optimization techniques," *Scientific Reports*, vol. 14, no. 1, p. 1337, 2024.

[8] T. Ali, H. U. Khan, F. K. Alarfaj, and M. Alreshoodi, "Hybrid deep learning and evolutionary algorithms for accurate cloud workload prediction," *Computing*, vol. 106, no. 12, pp. 3905–3944, 2024.

[9] J. Jeon, S. Park, B. Jeong, and Y.-S. Jeong, "Efficient container scheduling with hybrid deep learning model for improved service reliability in cloud computing," *IEEE Access*, vol. 12, pp. 65166–65177, 2024.

[10] M. Mayuranathan *et al.*, "An efficient optimal security system for intrusion detection in cloud computing environment using hybrid deep learning technique," *Advances in Engineering Software*, vol. 173, p. 103236, 2022.

[11] E. E. Sham and D. P. Vidyarthi, "Intelligent admission control manager for fog-integrated cloud: a hybrid machine learning approach," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 10, p. e6687, 2022.

[12] D. Saxena, J. Kumar, A. K. Singh, and S. Schmid, "Performance analysis of machine learning centered workload prediction models for cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 34, no. 4, pp. 1313–1330, 2023.

[13] D. Saxena and A. K. Singh, "OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments," *The Journal of Supercomputing*, vol. 78, no. 6, pp. 8003–8024, 2022.

[14] A. Attipalli *et al.*, "A deep-review based on predictive machine learning models in cloud frameworks for the performance management," Available at SSRN 5741282, 2022.

[15] A. Elkaradwy, A. Elshenawy, and H. Harb, "Enhancing cloud job failure prediction with a novel multilayer voting-based framework," *IEEE Access*, 2025.

[16] U. K. Lilhore *et al.*, "Cloud-edge hybrid deep learning framework for scalable IoT resource optimization," *J. Cloud Computing*, vol. 14, no. 1, p. 5, 2025.

[17] H. Zhou, "A novel approach to cloud resource management: hybrid machine learning and task scheduling," *J. Grid Computing*, vol. 21, no. 4, p. 68, 2023.