Deep Learning - Based Forecasting of Task Failures in Cloud Data Centers

Dr. P. Bharath Kumar Chowdary[1], Ameti Sadhana[1,2], Chintamaneni Mahalakshmi[1,3], Kamala Priya Vege[1,4], Kalakata Yagna Reddy[1,2*] and Srija Tulasi[1,5]

[1] Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering & Technology, Hyderabad, Telangana 500090, India
[2] JPMorgan Chase & Co., Hyderabad, Telangana 500081, India
[3] Experian Services India Private Limited, Hyderabad, Telangana 500081, India
[4] Providence Global Center India, Hyderabad, Telangana 500081, India
[5] Google India, Hyderabad, Telangana 500084, India

yagnareddy992@gmail.com

Abstract. A cloud data center empowers organizations to improve their performance, scalability and security by providing tools to store data and the infrastructure required to run applications. Its main objective is to provide high service reliability and service availability. But the lack of proper hardware and software resources may cause task and job failures which disrupt the services supported by the cloud data center. To recover to the pre-failure condition, many failures need a significant investment of time and resources.

An innovative failure prediction algorithm that harnesses the capabilities of multiple models, including the Hidden Markov Model (HMM), Hybrid CNN-LSTM (Long Short Term Memory), and Bi-LSTM (Bidirectional Long Short Term Memory) is proposed. This pioneering approach aims to proactively mitigate resource wastage in cloud environments by predicting task or job failures before they occur. Our ultimate goal is to categorize tasks as either successful or failed, and our research has shown that Bi-LSTM, among these models, emerges as the most promising choice, underscoring its potential to revolutionize task failure prediction in cloud computing.

Keywords: Task failure, Reliability, CNN-LSTM, Bi-LSTM, Hidden Markov Model

1 Introduction

A cloud data center's primary responsibility is to assure that the service will work according to the performance standards for a given amount of time and is available at promised times. Increased failure rates are reported even today in large-scale cloud data centers. The culprits of this are hardware component failures and system failures, which lead to task and job failures. These failures consume a huge amount of valuable time and resources to recover to the normal state. The cloud data centres, which contain

C. Kiran Mai et al. (eds.), *Proceedings of the Fourth International Conference on Advances in Computer Engineering and Communication Systems (ICACECS 2023)*, Atlantis Highlights in Computer Sciences 18,
https://doi.org/10.2991/978-94-6463-314-6_17

processors, memory units, disc drives, networking components, and other types of sensors, support the tasks that users have scheduled. The software will incur failures attributed to these many various forms of failures. In order to enhance the efficiency of failure recovery and continuing to execute the program, precise failure prediction is necessary in advance.

Therefore, a comparative analysis of performance among the different models chosen - Hybrid CNN-LSTM, Hidden Markov Model (HMM) and Bi-LSTM to predict breakdowns in tasks and jobs in cloud data facilities is presented.

2 Related Work

In research published by Mina Sedaghat et al., [1] the effect of random and associated failures on the reliability of workload in a data center is modeled. In order to determine task reliability in the presence of associated errors, it proposes an approximation method and statistical reliability model. It also addresses the problem of reliability-related task scheduling. In order to achieve the target reliability with the fewest additional jobs, it formulates the scheduling problem as an optimization problem. They used a scientific approach and performed cluster simulations with various error sources and dependability levels to check the algorithm's efficacy. Their findings demonstrate that the algorithm is capable of accurately approximating the minimal additional jobs required to ensure reliability.

A simulation model that employs a technique using neural networks to foresee hardware component breakdowns in data centres located in the cloud was proposed by Davis, Nickolas Allen, et al., [2]. RNN was used to analyze resource utilization patterns. With 89% accuracy, it can predict whether a host will fail before it really does, and with the removal of outliers. However, because different hosts exhibit different performance at various times, it is challenging to categorize a host's behavior.

Mohammad S. Jassas and others [3], used, DT, Random Forest, K-Nearest Neighbour, Extreme Gradient Boosting (XGBoost), Naive Bayes, Gradient Boost, and Quadratic Discriminant Analysis, and chose the best model by combining multiple evaluation criteria and selection techniques. The proposed model has a minimum recall accuracy of 95% and can recall up to 93.76% of failed jobs. The two traces did not produce the required findings because of the sparse amount of findings in Mustang and Trinity.

Tengku Asmawi et al., [4] suggested a paperfor the analysis of task failure assumptions in the cloud. They discovered that DT and RF performed remarkably better than others when the priority associated with the task or job was considered. According to their findings, the Logistic Regression model is reported more cost-effective which can handle any amount of data than the others. The Extreme Gradient Boosting classifier reports a 93.75% accuracy rating in predicting job failures while Random Forest modelling and decision tree techniques reported an accuracy rating of 88.95% for prediction of task failures. In order to save costs, it can be improved to consume less energy.

In this paper by Deepika Saxena et al., [5] virtual machines are placed in a failure tolerance unit that handles any failure beforehand while ensuring the accessibility of services to customers. When compared to other methods, service availability has grown

widely and the extent of working Virtual machines is scaled back by up to 34.27% and 82.97%, respectively. Proactive fault tolerance is included in the suggested approach. Combining reactive tolerance can strengthen the model.

Mohammed S Jassas et al.,[6] worked on a paper which was related to Failure Prediction Model. They used various methodologies like QDA, LDA, Linear Regression and for classification they used Random Forest classifier and XGboost classifier in their model. The model they proposed is trustable as well as interpretable. But they failed in overcoming challenges such as dealing with data skewness, hyperparameter-tuning and in-context evaluation.

Research on task resource consumption analysis and its relationship to failure prediction in the cloud was conducted by Rahul Sajjan et al. [7]. The main conclusions and contributions of this work are that whereas the resource consumption of completed tasks exhibits normal correlation, that of failed tasks exhibits high positive correlation. With 92.66% accuracy and 93.8% recall, synthetic minority oversampling and Extreme Gradient Boosting predicted the task state. However, SMOTE narrow margins the datasets used, necessitating the adoption of more advanced techniques to remedy the class imbalance issue.

Chunhong Liu and others, [8] experimented with SVM, OS-SVM, ELM, and OS-ELM as four different prediction models. They selected OS-ELM because it required less training time than the other models they considered. The incremental learning mechanism used in this method allowed for quick learning and strong adaptability. Better accuracy, job failure prediction, and resource conservation are not guaranteed by the suggested model.

Vamsi Krishna Bhandari [9] tried to calculate Proactive Fault Tolerance Through Cloud Failure Prediction using Machine Learning. To calculate fault tolerance in cloud machines he has used KNN, DT, Logistic Regression Analysis and SVM. This model has the ability to recognize certain message patterns that are associated with failure of tasks in data centers during failure training process. But this method fails to receive an impressive accuracy compared to other models.

3 Proposed Methodology

One method of foreseeing failures is to teach a computer to do so using communications or logs exchanged between various cloud components. The computer can recognise specific message patterns related to data centre failure during the training process. Later, the device can be used to determine whether or not a certain batch of message logs exhibits these patterns. This work seeks to focuses on building an algorithm that anticipates failure based on Bi-LSTM model to locate task and job failures in the cloud. The task failure prediction algorithm's objective is to determine whether tasks and jobs will fail or not based on a variety of factors

3.1 Preprocessing of Dataset

The initial approach is to apply preprocessing techniques on the Data Centre Workload Dataset. This dataset contains resource usage information of jobs in cloud hardware, each job/task having six attributes. The information in the dataset is of various jobs or tasks. Out of six attributes, three of them are related to access speed of cloud services. The last attribute for each job represents information whether the job will fail or not. Table 1. represents the various attributes present in the dataset. The training and testing datasets are converted from two-dimensional dataset to three-dimensional dataset using reshape, to feed as input to the deep learning model.

Table 1. Different attributes of the Dataset.

Name of attribute	Attribute Type	Attribute Description
job_id	float	Unique identifier for a job or task
memory_GB	float	Memory Usage
network_log10_MBps	float	Network speed
lcal_IO_log10_MBps	float	I/O transfer speed
NFS_IO_log10_MBs	float	Network File System transfer speed
status	int	Status of job or task

3.2 Proposed Classification Algorithm

Figure 1 represents the overall process of task failure prediction using our proposed classification algorithms. We have considered three algorithms- Hidden Markov Model (HMM), Bi-LSTM (Bi-directional Hybrid CNN-LSTM with Long Short Term Memory (LSTM). The first step is to collect data pertaining to jobs/tasks workload in cloud data centers. The next step is to preprocess the data collected using the data cleaning techniques of handling missing data. The preliminary processed information is next divided into train and test subsets. Model is trained separately using each of the mentioned deep learning algorithms for comparative analysis. Finally, the accuracy and F1-score of every algorithm are calculated and compared.

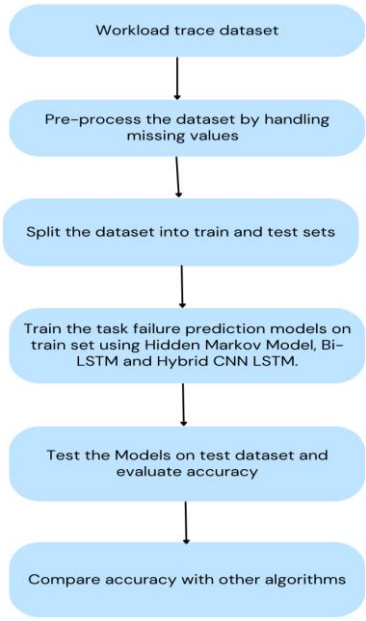

Fig. 1. Procedure of task failure prediction using proposed deep learning algorithms

Hidden Markov Model

A type of statistical modelling known as Hidden Markov models (HMMs) has been applied to data for accurate forecasting or decision-making. The system being modelled in HMM is a Markov process with hidden or unobserved states. With the use of the Markov assumption, it is a hidden variable model that can provide an observation of another hidden state. According to Markov's hypothesis, the hidden state is the following potential variable that cannot be viewed directly but can be inferred from viewing one or more states. A Markov process with hidden states is assumed to be the underlying process that produces the sequence of observations in an HMM, which is fundamentally a probabilistic model. The state transitions are governed by a transition probability matrix, and the emission probabilities are given by a set of output probability distributions. The algorithms for training and using HMMs make use of these assumptions to compute likelihoods, find the most likely order of hidden states and gauge the model's parameters from the data that has been collected.

Hybrid CNN-LSTM Algorithm

The second model in consideration is a hybrid CNN-LSTM model. Given that extremely long input sequences can be handled as blocks or subsections in the hybrid model because it includes both the CNN and LSTM models. For the purpose of training the hybrid model, additional subsequences of our sequential data are created for each sample.

There are multiple layers, or "multi building blocks," in the CNN architecture. The sequential data is first imported into a one-dimensional convolutional structure with 64 filters, two kernel sizes, and the ReLU activation function in order to create the output feature map. A pooling layer, which comes after that, reduces large feature maps in size to produce smaller feature maps. After that, a flattening layer converts the data into a one-dimensional array. The completely linked layer of the CNN model, which consists of 32 dense layers and the ReLU activation function, imports it. Finally, the output of the test prediction is executed by the final layer.

Bi-LSTM Algorithm

Bidirectional Long Short-Term Memory (BiLSTM) is a kind of recurrent neural network (RNN) utilized in applications involving sequence modelling. The architecture of BiLSTM extends the basic Long Short-Term Memory (LSTM) network by adding a second set of hidden states that process the input sequence in reverse order. This makes it possible for the network to capture information not only about the current input but also about the context from both past and future inputs. The architecture of BiLSTM consists of two LSTM layers that operate in opposite directions. The input sequence is fed into both layers simultaneously, with one layer processing the sequence from the beginning to the end and the other layer from the end to the beginning. The proposed algorithm

In the forward LSTM layer, the input sequence is processed one element at a time, and the hidden state and cell state are updated at each time step. In the backward LSTM layer, the input sequence will be handled in the reverse order, and the hidden state and cell state are updated accordingly. Each time step's output from the forward LSTM layer is combined with that time step's output from the backward LSTM layer. The final output is created by passing the produced outputs through a thick layer. The architecture of BiLSTM is especially helpful in modelling dependency over time in sequential data because it enables the capturing of both past and future context information. By considering context from both directions, the network can better capture dependencies that span across distant elements in the sequence. The proposed algorithm is given in table 2.

Table 2. Proposed Algorithm.

Algorithm	Bi - LSTM algorithm
Input	Data Centre Workload dataset
Output	Accuracy, F1-score
Step-1	Preprocessing (Handling missing data)
Step-2	Train and test sets are divided up in the dataset.
Step-3	To train the model, use the Bi - LSTM algorithm.
Step-4	Predict task status using the trained model
Step-5	Calculate classification parameters like accuracy, confusion matrix, f1-score etc
Step-6	Try comparing alternative models to the suggested model.

4 Results

This part discusses the task failure prediction and outcomes of the suggested strategy on the dataset that was discussed in the preceding sections. To make an informed conclusion about which model performs the best, we will consider three key performance metrics: accuracy, precision, and F1-score.

Accuracy: Accuracy measures the overall correctness and represents the ratio of correctly predicted instances to the total instances. Among the three models, 'Bi-LSTM' achieved the highest accuracy of 0.939, indicating that it correctly predicted task outcomes with the highest frequency compared to the other models.

Precision: Precision is a metric that focuses on the accuracy of positive predictions made by the model. In our evaluation, 'Bi-LSTM' exhibited the highest precision of 0.941. This means that when 'Bi-LSTM' predicts a task failure, it is often correct, minimizing the occurrence of false positives.

F1-Score: The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure that considers both false positives and false negatives. 'Bi-LSTM' achieved the highest F1-Score of 0.922, indicating that it maintains a strong balance between precision and recall.

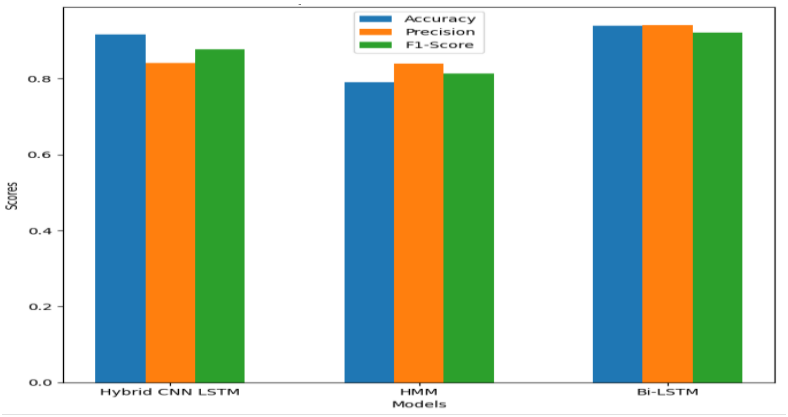

Fig. 2. Comparison of Models' Performance

The superior accuracy of Bi-LSTM as shown in Figure 2 can be attributed to its ability to capture the complex patterns and dependencies present in sequential data. by performing both forward and reverse processing on the input sequence and modeling both the temporal and contextual dependencies.

5 Conclusion

In cloud data centers, an application must meet a high standard of availability and dependability. For comparative analysis, we benchmarked our model against several well-

established methodologies, including Hybrid CNN-LSTM, Hidden Markov Models (HMM). The exceptional accuracy achieved by our Bi-LSTM model can be directly attributed to its proficiency in extracting and modeling complex patterns and dependencies embedded within sequential data. Through the dual processing approach examining the input sequence both forward and in reverse, and encompassing both temporal and contextual dependencies. Bi-LSTM harnesses the entirety of available information to make highly accurate predictions.

In summary, our approach leverages Bi-LSTM as a pioneering element, elevating the precision and novelty of our task failure prediction strategy. By seamlessly blending temporal and contextual awareness, Bi-LSTM not only showcases its superior predictive capabilities but also paves the way for more robust and insightful data-driven decision-making in cloud data center operations.

References

1. Mina Sedaghat, Eddie Wadbro (2016). DieHard: Reliable Scheduling to Survive Correlated Failures in Cloud Data Centers, IEEE Publishers, ISBN:978-1-5090-2453-7).
2. Davis, Nickolas Allen, et al. "Failuresim: a system for predicting hardware failures in cloud data centers using neural networks." 2017 IEEE 10th International Conference on Cloud Computing (CLOUD). IEEE, 2017.
3. Jassas, Mohammad S., and Qusay H. Mahmoud. "Analysis of Job Failure and Prediction Model for Cloud Computing Using Machine Learning." Sensors 22.5 (2022): 2035.
4. Tengku Asmawi, Tengku Nazmi, Azlan Ismail, and Jun Shen. "Cloud failure prediction based on traditional machine learning and deep learning." Journal of Cloud Computing 11.1 (2022): 1-19.
5. Saxena, Deepika, and Ashutosh Kumar Singh. "OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments." The Journal of Supercomputing 78.6 (2022): 8003-8024.
6. Jassas, Mohammad S., and Qusay H. Mahmoud. "A failure prediction model for large scale cloud applications using deep learning." 2021 IEEE International Systems Conference (SysCon). IEEE, 2021.
7. Shetty, Jyoti, Rahul Sajjan, and G. Shobha. "Task resource usage analysis and failure prediction in cloud." 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence). IEEE, 2019.
8. Liu, Chunhong, et al. "Predicting of job failure in compute cloud based on online extreme learning machine: A comparative study." IEEE Access 5 (2017): 9359-9368.
9. Bambharolia, Purvil, Prajeet Bhavsar, and Vivek Prasad. "Failure prediction and detection in cloud datacenters." International journal of scientific and technology research 6 (2017).
10. Pitakrat, Teerat, et al. "Hora: Architecture-aware online failure prediction." Journal of Systems and Software 137 (2018): 669-685.