IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 3, March 2024

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

Task Failure Prediction in Cloud Data Centers Using Deep Learning

Prasanth Garneedi, Raj Kamal Kollipara, Gopi Reddy Kurri, Maheswar Mane

U.G. Students, Department of Computer Science Engineering (CIC Specialisation), Vasireddy Venkatadri Institute of Technology, Nambur, Andhra Pradesh, India

ABSTRACT: A large-scale cloud data centre needs to provide high service reliability and availability with low failure occurrence probability. However, current large-scale cloud data centers still face high failure rates due to many reasons such as hardware and software failures, which often result in task failures. Such failures can severely reduce the reliability of cloud services and also occupy huge amount of resources to recover the service from failures. Therefore, it is important to predict task failures before occurrence with high accuracy to avoid unexpected wastage. Many machine learning and deep learning based methods have been proposed for the task failure prediction by analyzing past system message logs and identifying the relationship between the data and the failures. In order to further improve the failure prediction accuracy of the previous machine learning and deep learning based methods, in this project, we propose a failure prediction algorithm based on multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) to identify task failures in the cloud. The goal of Bi-LSTM prediction algorithm is to predict whether the tasks are failed or completed.

KEYWORDS: Deep learning, Bi-LSTM, Machine Learning, Cloud Data Centers, System log analysis, Neural Network Architecture, Task Failures, Prediction

1. INTRODUCTION

Cloud computing has revolutionized the way modern businesses operate by providing scalable, on-demand access to a wide range of computing resources. Large-scale cloud data centers play a pivotal role in this ecosystem, serving as the backbone infrastructure for hosting a myriad of applications and services. However, ensuring high service reliability and availability in such environments remains a critical challenge. Despite significant advancements in hardware and software technologies, cloud data centers still face high failure rates, resulting in service disruptions and resource wastage.

Task failures within cloud data centers can stem from various sources, including hardware malfunctions, software bugs, and resource contention. These failures not only undermine the reliability of cloud services but also entail substantial operational overheads for recovery and maintenance. Consequently, there is a pressing need for effective mechanisms to predict task failures proactively and mitigate their adverse impact on service performance.

Machine learning and deep learning techniques have emerged as promising approaches for task failure prediction in cloud environments. By leveraging historical system logs and analyzing patterns in the data, these methods aim to discern the underlying factors contributing to failures and anticipate their occurrence with greater accuracy. However, achieving robust and precise predictions remains a challenge, particularly in dynamic and complex cloud environments.

In this context, our research introduces a novel failure prediction algorithm based on multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) neural networks. Building upon the strengths of deep learning architectures, Bi-LSTM models are capable of capturing intricate temporal dependencies in sequential data, making them well-suited for analyzing system logs and identifying precursors to task failures. By harnessing the power of Bi-LSTM networks, our algorithm seeks to enhance the predictive capabilities for detecting impending task failures in cloud data centers.

The primary objective of our study is to develop an advanced failure prediction framework that can accurately discriminate between successful and failed tasks in real-time. Through empirical evaluations and comparative analyses, we aim to demonstrate the efficacy and reliability of our proposed Bi-LSTM-based approach in addressing the challenges associated with task failure prediction in cloud computing environments. Ultimately, our research endeavors to contribute to the advancement of proactive fault management strategies and the overall resilience of cloud-based services.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

II. LITERATURE REVIEW

Prior research in the field of failure analysis and prediction in cloud data centers has investigated various methods for understanding and anticipating system failures. This section presents a review of relevant studies, which are classified into two categories: failure analysis in cloud data centers and failure prediction methods.

Failure Analysis in Cloud Data Centers*:*

- Ford et al. [3] conducted a study on the impact of correlated failures on the availability of distributed storage systems in Google clusters. Their findings shed light on the challenges posed by correlated failures in large-scale cloud environments.

Failure Prediction Methods:

- Zhao et al. [4] utilized Hidden Markov Models (HMM) and Hidden Semi-Markov Models to predict disk failures in cloud storage systems, highlighting the importance of probabilistic modeling in failure prediction.
- Pitakrat et al. [5] proposed a hierarchical online failure prediction approach named Hora, which combines failure propagation modeling and Bayesian network-based techniques for software system failure prediction.
- Zhang et al. [6] introduced PreFix, a tool based on Random Forest (RF), for accurately predicting switch failures in cloud networks, demonstrating the efficacy of ensemble learning methods in failure prediction.
- Das et al. [7] developed a powerful technique using Long Short-Term Memory (LSTM) networks to process High-Performance Computing (HPC) logs for efficient failure prediction. Their approach employs a three-phase deep learning strategy to recognize log event chains leading to failures and predict lead times.
- Du et al. [8] proposed DeepLog, a framework for predicting task and job failures in Hadoop distributed file systems using LSTM and density clustering methods. Their approach effectively models log sequences and groups similar events to improve prediction accuracy.
- Xu et al. [9] introduced an RNN-based model for predicting hard disk drive failure and assessing health degrees using SMART features. Their model captures long-term dependencies in time-series data to make accurate predictions about future disk health.

III. PROPOSED METHODOLOGY

Overview of the Approach:

In this study, we developed a task failure prediction model using a multi-layer Bidirectional Long Short-Term Memory (Bi-LSTM) neural network. The Bi-LSTM architecture was chosen for its ability to capture both forward and backward temporal dependencies in sequential data, making it well-suited for analyzing time-series data such as system logs in cloud data centers. By leveraging the capabilities of Bi-LSTM networks, we aimed to accurately predict task failures in cloud environments, thereby improving the reliability and availability of cloud services.

Data Collection:

The dataset used for training and testing the Bi-LSTM model was obtained from Kaggle. It consists of various attributes related to cloud task executions, including CPU usage, memory usage, page cache, disk usage, priority, task resubmissions, scheduling delay, event types, and task outcomes (failed or not failed). Initially, the dataset underwent preprocessing steps to clean and prepare it for analysis.

- Feature Engineering: We calculated additional features such as the mean input/output time difference (meanio) by subtracting the output time from the input time. This feature was deemed important for capturing time-related patterns in task executions.
- Handling Missing Values: We observed missing values primarily in the 'diskusage' and 'scheduling delay' columns. Given the relatively small proportion of missing values compared to the dataset size, we chose to drop these rows.
- Data Transformation: The 'cpu usage and memory usage' column contained string representations of CPU and memory usage. We split this column into two separate columns ('cu' for CPU usage and 'mu' for memory usage) and converted the string values to float type for further analysis.
- Label Encoding: The 'event' column, representing different types of events associated with task executions, was encoded using label encoding to convert categorical values into numerical representations.
- Normalization: Finally, we performed min-max scaling on the dataset using the MinMaxScaler to normalize the features to a common scale, which is essential for neural network training.

The preprocessed dataset, denoted as 'data2.csv,' was then used for model training and evaluation.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

IV. IMPLEMENTATION

Model Architecture:
An example of the architecture of Bi-LSTM model is shown in Fig 1, which has an input layer, two Bi-LSTM layers, an output layer, and a Logistic Regression (LR) layer to determine whether a task or job has been successful or failed.

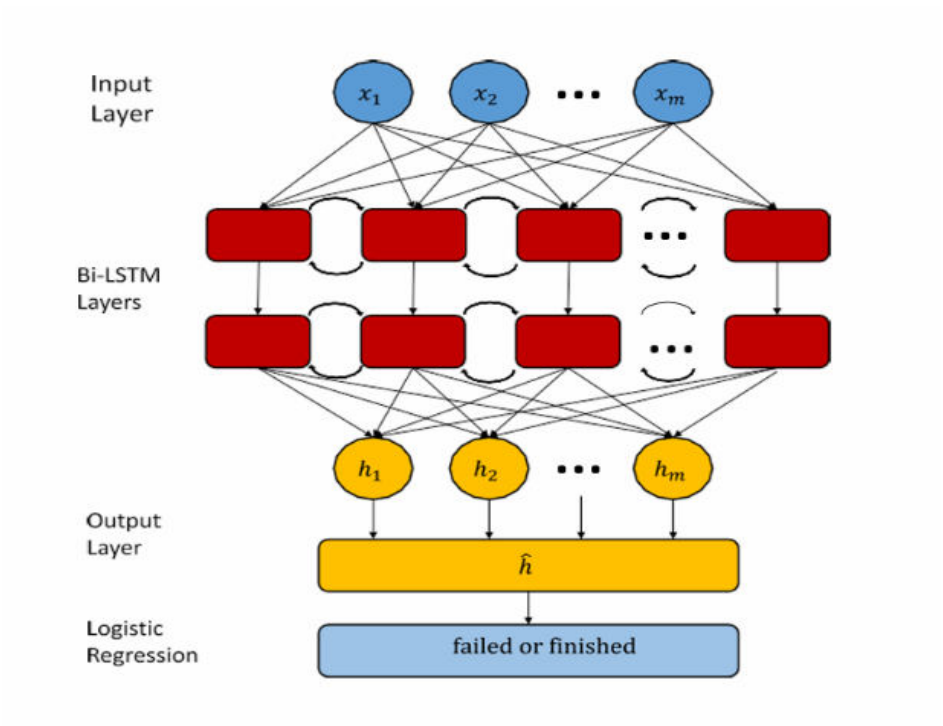


Fig 1: Architecture of multi-layer Bi directional LSTM(Bi-LSTM)

Input Layer:
The input layer of a bidirectional LSTM (Bi-LSTM) model is responsible for receiving the input data and passing it to the subsequent layers for further processing. The input layer is composed of a sequence of neurons, each of which corresponds to one element of the input sequence. The number of neurons in the input layer would correspond to the length of the input sequence, and the dimensionality of the input data, such as the number of features, would determine the size of each neuron in the input layer.

Bi LSTM Layer:
The bidirectional LSTM (Bi-LSTM) layer in the Bi-LSTM model has two parts as shown in Fig 2, the forward state and the backward state. Each sequence of inputs is presented to both the forward and backward states, which allows the model to capture information from both directions and hence better capture the context and dependencies within the sequence. The forward state processes the sequence from the beginning to the end, while the backward state processes the sequence from the end to the beginning. The outputs from both states are concatenated to produce the final output of the Bi_LSTM layer.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

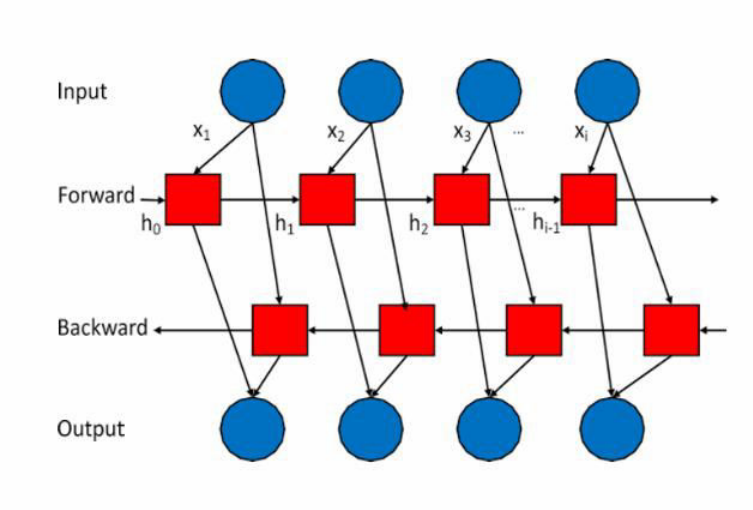


Fig 2: Structure of Bi-LSTM Layer

Each memory cell has three gates to protect and control its state as shown in Fig 3, including the input gate, forget gate and output gate. The input gate determines which cell state should be updated. The forget gate decides what information should be overlooked. The output gate resolves which part of the cell state will be exported.

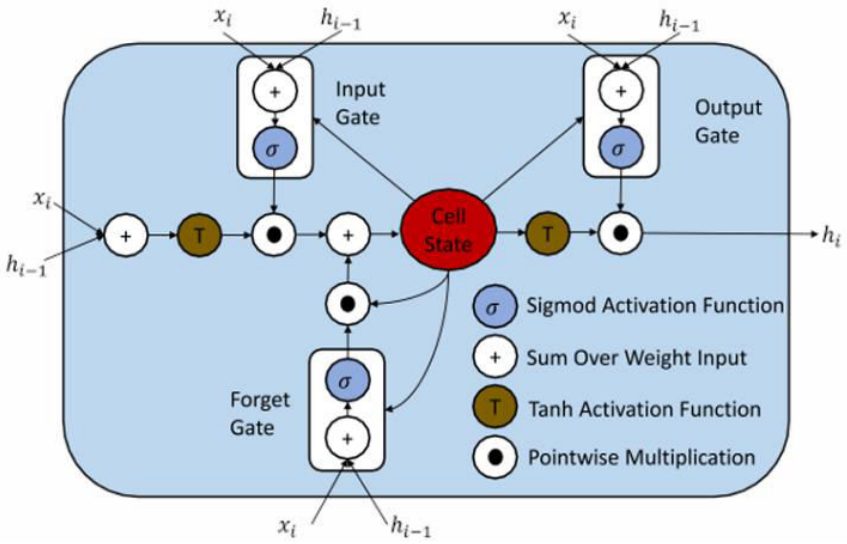


Fig 3: Structure of memory cell in Bi-LSTM

Output Layer:
In the output layer, from an input sequence {x1, x2 ... xm}, the memory cells in the Bi-LSTM layers will produce a representation sequence {h1, h2 ... hm}. We define the mean of these outputs as the mean pooling.

Logistic Regression Layer:
The output of the Bi-LSTM layer is fed to the logistic regression layer, which uses a logistic regression function to classify the input sequence as either failed or finished. Setting up a threshold for the probability value of failure is a common approach in binary classification problems. The threshold is a value between 0 and 1 that determines the level of confidence required for the model to classify an instance as belonging to a particular class.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

Model Training:
The model was trained using the training dataset with an Adam optimizer and binary crossentropy loss function. During training, the input data was reshaped to meet the requirements of the LSTM input shape. The training process involved 50 epochs with a batch size of 75. Dropout layers were incorporated to prevent overfitting and enhance model generalization.

Performance Metrics:
To illustrate the performance of our method, we use accuracy as the metrics to determine the results. In a confusion matrix, accuracy is defined as the ratio of correctly classified instances to the total number of instances. Mathematically, it can be expressed as:

$$\text{Accuracy} = (TP+ TN) / (TP+TN+FP+FN)$$
$$\text{Precision} = (TP)/ (TP+FP)$$
$$\text{Recall} = \text{True Positive (TP)} / \text{True Positive (TP)} + \text{False Negative (FN)}$$

where true positives (TP) are the number of instances correctly predicted as positive. true negatives (TN) are the number of instances correctly predicted as negative. false positives (FP) are the number of instances predicted as positive but actually negative. false negatives (FN) are the number of instances predicted as negative but actually positive.

Model Evaluation:
The trained model was evaluated using the testing dataset to assess its predictive performance. Evaluation metrics such as accuracy_score were utilized to measure the model's ability to correctly classify task outcomes. The testing process revealed accuracy of 98%, indicating the effectiveness of the Bi-LSTM model in predicting task failures in cloud data centers.

Model Deployment with Streamlit:
To facilitate real-time predictions of task failure, our trained Bidirectional Long Short-Term Memory (Bi-LSTM) model was integrated into a user-friendly web application using Streamlit - a Python library for building interactive web applications. The integration process involved the following steps:

- The application interface was designed using Streamlit's simple and intuitive components, enabling users to input task parameters conveniently.
- Key parameters such as CPU usage (cu), memory usage (mu), page cache (pagecache), disk usage (diskusage), task priority (priority), scheduling delay (scheduling delay), and mean I/O (meanio) were provided as input fields.
- The trained Bi-LSTM model, stored in the Hierarchical Data Format (HDF5) file format, was loaded into the application using TensorFlow's load_model function. This ensured seamless access to the model for making predictions.
- Upon submitting the input parameters through the form, the user-triggered event invoked the prediction process.
- The entered data was preprocessed and formatted to match the input requirements of the Bi-LSTM model.
- The model made predictions based on the provided input data.
- The predicted task status—either "Failed" or "Succeeded"—was displayed to the user along with the corresponding input values.

This integration allowed cloud administrators or users to conveniently input task parameters and obtain real-time predictions regarding task outcomes directly through a web browser, enhancing operational efficiency and decision-making in cloud environments.

Technologies Used:
The programs and technologies crucial to the realization of our project were:

- Python : Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.
- Keras and Tensorflow: TensorFlow is an open-sourced end-to-end platform, a library for multiple machine learning tasks, while Keras is a high-level 7 neural network library that runs on top of TensorFlow.
- Google Collab: Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

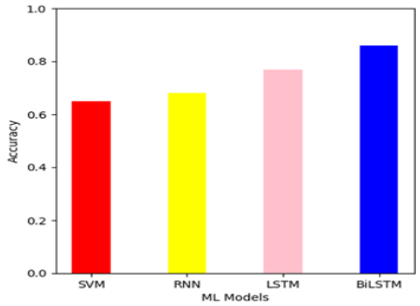
| DOI:10.15680/IJIRSET.2024.1303164 |

- Google Drive: Along with colab google drive was used to manage and save all the files such as model weights, model structure, image dataset, etc.
- Pandas: Pandas is a popular open-source library for data manipulation and analysis in Python.
- Sklearn: Scikit-learn (also known as sklearn) is a popular open-source library for machine learning in Python. It provides a wide range of tools for supervised and unsupervised learning, including classification, regression, clustering, and dimensionality reduction.

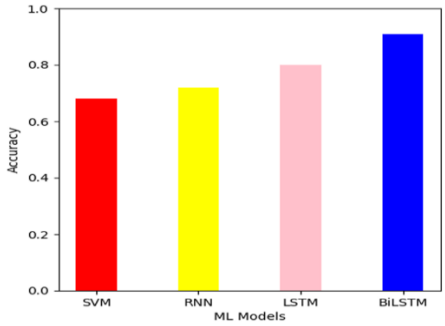
V. RESULTS

We classify the termination statuses of task submissions based on the attributes and performance data. In all the target classes, the status finish is considered as one class, and the status failed is considered as other class. 1. We took only 1000 tasks and accuracy of all the models were checked and results were obtained.

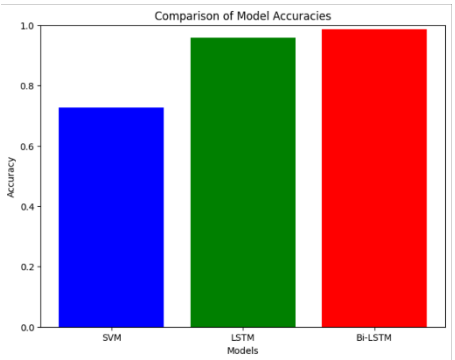
SVM: 65%, RNN: 68%, LSTM: 77%, Bi-LSTM: 86%



We took 20000 tasks and accuracy of all the models were checked and results were obtained. SVM: 68%, RNN: 72%, LSTM: 80%, Bi-LSTM: 91%



We 280896 (entire dataset) tasks and accuracy of all the models were checked and results were obtained. SVM: 72% LSTM: 95% Bi-LSTM: 98%

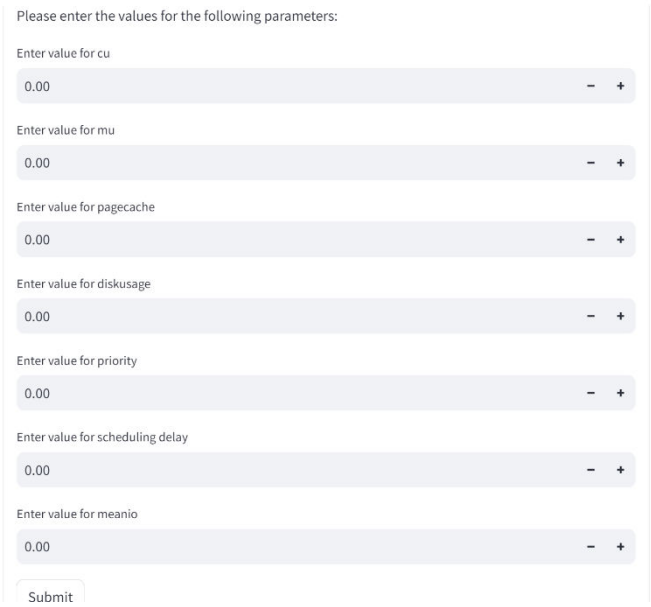


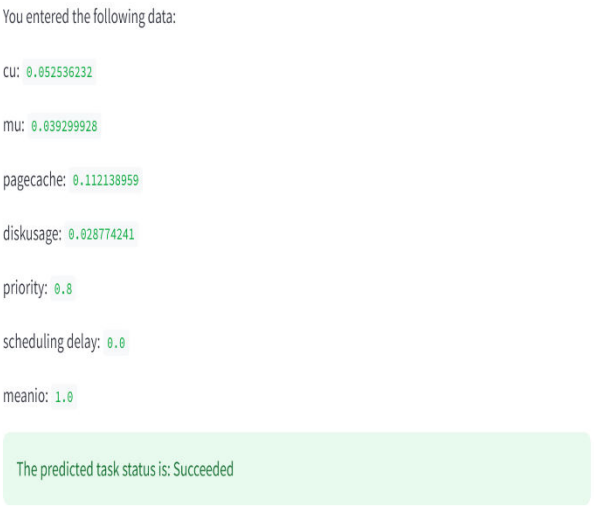
|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

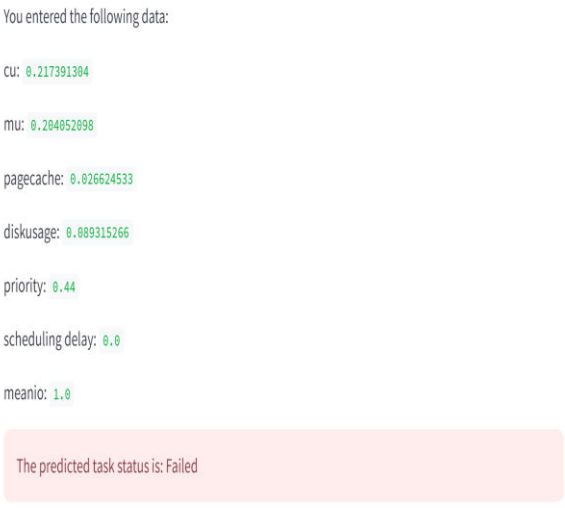
|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

Streamlit Web Interface:
The Streamlit web interface serves as a user-friendly platform for cloud administrators to predict task outcomes based on input parameters. Upon submission of the input parameters, the interface dynamically processes the data and provides a prediction indicating whether the task is predicted to fail or succeed.







We classify the termination statuses of task submissions based on the attributes and performance data. In all the target classes, the status finish is considered as one class, and the status failed is considered as other class.
The result follows :

SVM < RNN < LSTM < Bi-LSTM.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303164 |

SVM model shows the least accuracy while Bi-LSTM model shows the highest accuracy . Performance of SVM is worse than other models because SVM only has better performance when the dataset is not so big. The reason for not so high accuracy of RNN is the data with long term dependancies. LSTM cannot adjust the weight of further data point for a given time in the time series dataset. For Bi-LSTM, it has the forward and backward states which can more accurately determine the weights of data items that are closer and further to the given time in the prediction. Hence, Bi LSTM shows highest accuracy.

VI. CONCLUSION & FUTURE SCOPE

Bi-LSTM has shown better performance in predicting task and job failures compared to previous methods. It can capture the long-term dependencies in the time series data and also considers the data in both forward and backward directions, which helps to improve the accuracy of the predictions. We have compared Bi-LSTM with other comparison methods including statistical, machine learning and deep learning based methods and evaluate the performance. The results show that we achieved 98% percent accuracy in task failure prediction .

In the future work, we will first try to implement our methods in real data center to examine the real-world performance and then work on how to use the failure prediction results from Bi-LSTM to build a failure recovery system in data center which can further improve the fault tolerance performance. We will also implement a job scheduling algorithm that can migrate the tasks and jobs to a suitable machine. The scheduler will be scalable and have the capability of dynamically adjusting its scheduling decisions based on the Bi-LSTM prediction results.

REFERENCES

[1] Overcoming the cause of data center outrages, 2019. Accessed: Apr. 2019. [Online]. Available:https://www.365datacenters. com/portfolio-items/overcoming causes-data-center-outages/
[2] 2015. Accessed: Apr. 2019. [Online]. Available: https:// techcrunch.com/2015/01/02/following-bing-coms-briefoutage search-yah oo-com goes-down-too/
[3] D. Ford et al., “Availability in globally distributed storage systems,” in Proc. USENIX Symp. Operating Syst. Des. Implementation, 2010, pp. 61–74.
[4] Y. Zhao, X. Liu, S. Gan, and W. Zheng, “Predicting disk failures with HMM-and HSMM-based approaches,” in Proc. Ind. Conf. Data Mining, 2010, pp. 390–404.
[5] T. Pitakrat, D. Okanovic, A. van Hoorn, and L. Grunske, “Hora: Architecture aware online failure prediction,” J. Syst. Softw., vol. 137, pp. 669–685, 2018.
[6] S. Zhang et al., “PreFix: Switch failure prediction in datacenter networks,” Proc. ACM Meas. Anal. Comput. Syst., vol. 2, 2018, Art. no. 2.
[7] A. Das, F. Mueller, C. Siegel, and A. Vishnu, “Desh: Deep learning for system health prediction of lead times to failure in HPC,” in Proc. 27th Int. Symp. High Perform. Parallel Distrib. Comput., 2018, pp. 40–51.
[8] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly detection and diagnosis from system logs through deep learning,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2017, pp. 1285–1298. 23
[9] C. Xu, G. Wang, X. Liu, D. Guo, and T. Liu, “Health status assess ment and failure prediction for hard drives with recurrent neural networks,” IEEE Trans. Comput., vol. 65, no. 11, pp. 3502–3508, Nov. 2016
[10] Y. Cheng, H. Zhu, J. Wu, and X. Shao, “Machine health monitoring using adaptive kernel spectral clustering and deep long short term memory recurrent neural networks,” IEEE Trans. Ind. Informat., vol. 15, no. 2, pp. 987–997, Feb. 2019.
[11] P. Zhou et al., “Attention-based bidirectional long short-term memory networks for relation classification,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, 2016, pp. 207–212.
[12] Z. Huang, W. Xu, and K Yu, “Bidirectional LSTM-CRF models for sequence tagging,” 2015, arXiv:1508.01991.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details