Juni Khyat
(UGC Care Group I Listed Journal)
ISSN: 2278-4632
Vol-14, Issue-2, Febbruary: 2024

Deep Learning-based Anomaly Detection for Cloud Service Tasks

Vinitha Reddy
Master of Computer Application
Chaitanya Bharathi Institute of Technology(A),
Hyderabad, Telangana, India
Vinithareddie207@gmail.com

Dr.B.Indira
Master of Computer Application
Chaitanya Bharathi Institute of Technology(A),
Hyderabad, Telangana, India

***Abstract*- Cloud data centers are becoming increasingly important for running critical applications and services. However, failures in cloud data centers can have severe consequences, including service downtime and financial losses. To mitigate these risks, predicting task failures in cloud data centers has become an important research topic. In this project, we propose a deep learning-based approach for task failure prediction in cloud data centers. Specifically, we utilize a long short-term memory (LSTM) neural network and Bi-LSTM to model the temporal dependencies of the task execution data. We also introduce a novel feature extraction method that combines the task execution history and resource utilization information to enhance the prediction accuracy. We will apply RF, DT, CNN, CNN+LSTM is used for feature values and Bi directional Long Short Term Memory (Bi LSTM) is used to predict whether the tasks and jobs are failed or completed. With the Voting Classifier we will build the model which will be used for predicting the result. Our results show that deep learning-based approaches can be effective for task failure prediction in cloud data centers, and our proposed method can provide valuable insights for improving the reliability and availability of cloud services.**

Keywords – Cloud Data Center, Task Failure prediction, cloud services, reliability, Random Forest, Decision Tree and Deep learning.

I INTRODUCTION

Cloud computing is a popular service nowadays because it delivers on-demand services, resource savings, and high reliability. The cloud data center, which have processors, memory units, disc drives, networking equipment, and other types of sensors, support a large number of user applications (i.e., jobs). Users can make requests to the cloud for the execution of apps and the storing of data. Physical machines (PMs) make up each cloud data center, and each PM is capable of supporting a group of virtual machines (VMs). Each VM processes the tasks that the users send it. Such a sizable cloud data center may house hundreds of thousands of computers, many of which often operate many apps and get work requests from people all over the world every second. With such diverse workloads and heterogeneity, a cloud data center may occasionally be susceptible to various failure types (such as disc, software, and hardware problems). Consider a software failure: in January 2015, Yahoo Inc. and Microsoft's Bing search engine collapsed for 20 minutes, costing nearly $9,000 per minute to restart.

Previous studies shown that a significant cause of cloud service disruptions is hardware failure, particularly disc loss. The program will experience failures due to these several distinct failure kinds. Therefore, reliable application failure prediction can increase the effectiveness of recovering from failures and keeping applications functioning.

I LITERATURE REVIEW

Eddie Wadbro et al. [1] focused on the impact of correlated failures in large-scale data centers on job reliability. It addresses failures caused by power outages or network component issues, affecting multiple physical machines and their tasks simultaneously. The study presents a statistical reliability model and an approximation technique to compute job reliability in the presence of such correlated failures. Additionally, the paper formulates a scheduling problem as an optimization task to achieve desired reliability with minimal extra tasks and proposes an efficient scheduling algorithm for this purpose.

Thanyalak Chalermarrewong et al. [2] introduced a framework for online failure prediction in data centers, aiming to address the high failure rate and potential compromises in system performance. The focus is on hardware failure prediction to ensure graceful handling of failures in data centers with long-running applications and intensive workloads. Two methods, ARMA and Fault Tree Analysis, are employed for prediction, and experiments on a simulated cluster show a high prediction accuracy of 97%. The paper concludes that the proposed framework is practical and holds potential for future adaptation in real data center environments.

Haoyu Wang et al. [3] In modern cloud data centers, cascading failures can lead to numerous Service Level Objective (SLO) violations. Cascading failures occur when a group of physical machines in a failure domain fails, causing their workloads to shift to another domain. Existing methods have limited effectiveness in handling such cascading failures. To address this issue, the paper proposed the Cascading Failure Resilience System (CFRS) comprising three methods: Overload-Avoidance VM Reassignment (OAVR), VM backup set placement (VMset), and Dynamic Oversubscription Ratio Adjustment (DOA). Trace-driven simulations demonstrate that CFRS outperforms other comparative methods, reducing the number of domain failures, failed PMs, and SLO violations.

Haiying Shen et al. [4] has addressed the issue of network latency caused by incast congestion in data centers due to a massive influx of requests to the front-end server simultaneously. Existing solutions for incast problems lack proactive measures. To overcome this, the paper introduced the Proactive Incast Congestion Control system (PICC). PICC limits the number of data servers concurrently connected to the front-end server through intelligent data placement. Additionally, PICC employs a queuing delay reduction algorithm to prioritize data objects with smaller sizes and longer queuing times, further improving performance.

Jiechao Gao et al. [5] focused on improving the reliability and availability of a large-scale cloud data center by predicting task and job failures with high accuracy. The current data centers face high failure rates due to various reasons, impacting service reliability and resource usage. To address this, the proposed approach utilizes a multi-layer Bidirectional Long Short Term Memory (Bi-LSTM) algorithm to predict task and job failures by analyzing past system message logs. The goal is to determine whether tasks and jobs will fail or complete. The trace-driven experiments demonstrate that the Bi LSTM algorithm outperforms other state-of-the art prediction methods, achieving 93% accuracy for task failure prediction and 87% accuracy for job failure prediction.

Avinab Marahatta et al. [6] proposed an AI-driven energy-aware proactive fault-tolerant scheduling scheme for cloud data centers (CDCs). Task failure is common in CDCs due to complex data stream computation and task dependencies, leading to poor user experience and increased energy consumption. The scheme includes a prediction model based on machine learning to classify tasks as "failure-prone" or "non-failure-prone" based on predicted failure rates. Two efficient scheduling mechanisms are then employed to allocate these tasks appropriately to hosts in the CDC. Evaluation results demonstrate that this scheme intelligently predicts task failure, achieves better fault tolerance, and reduces total energy consumption compared to existing schemes.

Jyothi Shetty et al. [7] focused on improving the reliability of cloud computing systems through failure prediction. It conducts a statistical analysis of resource usage data from tasks in the large Google cluster dataset to understand failure characteristics. The study reveals variations in resource usage patterns, execution duration, and resource consumption between failed and finished tasks. With Synthetic Minority Oversampling Technique (SMOTE) and XGboost, the proposed approach achieves a high precision of 92% and recall of 94.8% in predicting task failures, despite dealing with a highly imbalanced dataset.

Jomar Domingos et al. [8] developed a new methodology for failure prediction in cloud applications using ensemble machine learning. The approach involves identifying system state patterns preceding failures (symptom detection) by training different models with failure datasets obtained through realistic failure injection. These ensembles are then validated using fault injection. The ability to predict failures and take preventive measures before their occurrence is crucial for critical application scenarios in cloud computing, making ensemble-based machine learning models a promising approach for achieving this goal.

Mohammad Jassas et al. [9] focused on failure analysis in public and private cloud providers to understand the causes of different failures and find solutions. The main objective is to enhance understanding of job failure in cloud computing environments. The study reveals a correlation between failed jobs and requested resources like memory, CPU, and disk space, suggesting various techniques to improve cloud application reliability and availability, including scheduling algorithms, job failure prediction, task resubmission limits, and priority policy changes.

Yanwen Xie et al. [10] addressed the challenge of making accurate failure predictions for various disk models in a heterogeneous data center. The proposed OME (Optimized Modeling Engine) builds a basis predictive model with one-for-all modeling and then optimizes predictions for each disk model using one-for-one and transfer learning modeling. OME achieves automation through simple but effective transfer learning, cross-validation, tuning space pruning, and parallelism using a directed acyclic graph. Evaluation on real-world data shows that OME outperforms previous one-for-all predictive models by 18.5% overall, with improvements of over 30% for 43.3% of the disk models.

I METHODOLOGY

A. *Proposed System:*

It offers on-demand services, resource savings, and high reliability, cloud computing is a widely used service today. Many applications (i.e., jobs) from users are supported by the cloud data centers, which contain processors, memory units, disk drives, networking devices, and many sorts of sensors. Users can ask the cloud to store data and operate apps by sending requests in this manner. Physical machines (PMs) make up each cloud data center, and each PM is capable of supporting a group of virtual machines (VMs). Each VM processes the tasks that are sent by users. Such a sizable cloud data center can house tens of thousands of servers, many of which operate numerous applications and get work requests from people all over the world every second.

B. *Advantages of the proposed system*

- Detects task failures and job failures with high accuracy.

- Observed that the time cost overhead for Bi LSTM is almost the same compared with RNN and LSTM, which means Bi-LSTM can achieve higher prediction performance with no further time cost.

C. *Modules*

1) *Data Collection :* This function is responsible for gathering data from various sources within the cloud data center. It may include collecting information on task execution history, resource usage, system logs, hardware health, network statistics, and any other relevant metrics. The data collected will serve as the input for training the deep learning model.

2) *Data Preprocessing :* The data preprocessing function will handle tasks such as data cleaning, normalization, feature scaling, and handling missing values to ensure the data is in a suitable format for training the deep learning model.

3) *Feature Selection:* In this function, relevant features that contribute significantly to task failure prediction will be selected. The function will perform feature selection techniques, such as correlation analysis, feature importance ranking, or dimensionality reduction, to identify the most informative features to be used in the model.

4) *Model Selection:* For the task failure prediction, this function selects the best deep learning architecture. Convolutional neural networks (CNNs), long short-term memory (LSTM) networks, recurrent neural networks (RNNs), and hybrid models will all be considered, and the model that best fits the given situation will be chosen.

5) *Model Training:* The model training function takes the preprocessed data and the selected deep learning architecture and trains the predictive model. It involves setting hyperparameters, using optimization techniques (e.g., SGD), and executing the backpropagation algorithm to update the model's weights and biases.

6) *Model Evaluation:* After the model has been trained, it must be assessed to see how well it performed. To assess the model's capability to properly forecast task failures, the model evaluation function will employ suitable evaluation measures including accuracy, precision, recall, F1-score, and ROC curves.

7) *Real-time Monitoring*: Once the model is trained and evaluated, it needs to be deployed in real-time to continuously monitor ongoing tasks in the cloud data center. This function will be responsible for the real-time implementation of the predictive model, generating alerts or notifications when it predicts an impending task failure.

8) *Fine-Tuning and Optimization*: This function will continuously monitor the model's performance in a real-world setting. If necessary, it will perform fine tuning and optimization to improve the model's accuracy and adapt it to changing conditions in the cloud data center.

9) *Reporting and Visualization*: To make the insights more accessible and understandable to cloud data center operators, a reporting and visualization function can be implemented. It will generate informative visualizations and reports about the model's predictions, performance, and trends.

10) *Feedback and Retraining*: As the cloud data center environment evolves, new data will be generated. The feedback and retraining function will enable the system to periodically update the model with new data to maintain its accuracy and effectiveness over time.

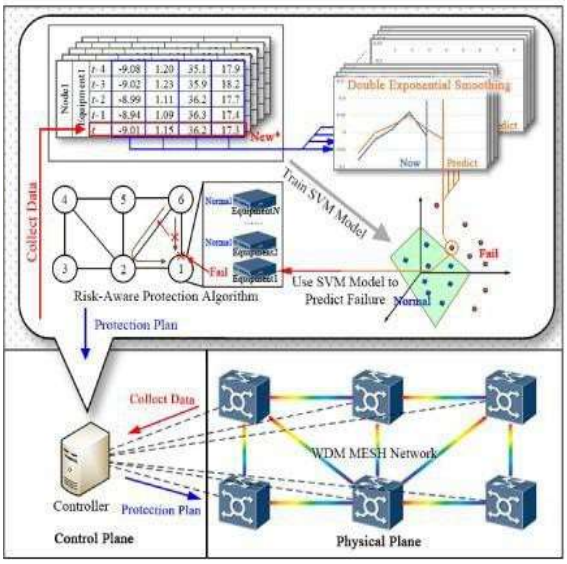

Fig. 1 Project Architecture

IV. IMPLEMENTATION

A. *ALGORITHMS:*

1) Random Forest: A popular algorithm for supervised machine learning used to solve classification and regression issues is random forest. On various samples, it constructs decision trees and uses their average for classification and majority vote for regression.

2) *Decision Tree*: To decide whether to divide a node into two or more sub-nodes, decision trees employ a variety of techniques. The homogeneity of newly formed sub-nodes is increased by sub-node formation. In other words, we may claim that the node's purity improves in relation to the desired variable.

3) *KNN:* K Nearest Neighbour is a straightforward algorithm that sorts incoming information or instances based on a similarity metric after storing all of the existing examples. It is mostly utilised to categorise

data points according to their neighbors are classified

4) *Voting Classifier*: A voting classifier is a machine learning estimator that trains numerous base models or estimators and predicts by aggregating the results of each base estimator. Aggregating criteria can be coupled voting decisions for each estimator output.

5) *Support Vector Machine:* Support Vector Machine (SVM) is a supervised machine learning technique that may be used for both regression and classification. Although we often refer to regression concerns, categorization is the most appropriate term. Finding a hyperplane in an N-dimensional space that clearly classifies the data points is the goal of the SVM method.

6) *CNN:* For deep learning algorithms, a CNN is a unique kind of network architecture that is utilised for pixel-intensive tasks like image recognition. CNNs are the ideal network architecture for recognising and detecting objects in deep learning, even if there are other types of neural networks available.

7) *CNN+LSTM:* In the CNN LSTM architecture, Convolutional Neural Network (CNN) layers and LSTMs are linked to extract features from input data.

8) *LSTM:* Long short-term memory (LSTM) is a type of artificial neural network used in artificial intelligence and deep learning. Unlike traditional feedforward neural networks, LSTM has feedback connections. A recurrent neural network (RNN) of this type can analysenot just single data points (such as pictures), but also complete data sequences (such as audio or video).

9) *BiLSTM:* BiLSTM stands for bidirectional long-term memory. Future data is frequently disregarded by LSTM while processing time series in general. On the basis of LSTM, BiLSTM connects the two hidden layers by processing series data in both forward and backward orientations.

10) *RNN:* A recurrent neural network (RNN) is a type of artificial neural network in which connections between nodes can form a cycle, allowing the output of some nodes to influence the input received by other nodes in the same network. It can display temporal dynamic behaviour as a result of this. RNNs, which are derived from feedforward neural networks, may process input sequences of different lengths by using their internal state (memory). They may therefore be used for tasks like connected, unsegmented handwriting recognition or speech recognition.

V. EXPERIMENTAL RESULTS

Screenshots

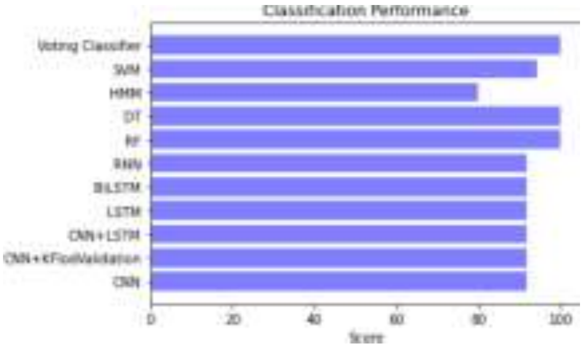

Fig. 6 Accuracy Result

VI. CONCLUSION

In cloud data centers, high service reliability and availability are crucial to application. We proposed a failure prediction model to accurately predict the termination statuses of tasks and jobs. When compared to prior approaches, RF can more reliably predict the termination states of tasks. In order to modify the weight of both closer and farther input characteristics, we first input the data into forward and backward states in our approach. We then discover that additional input characteristics are critical to getting high prediction accuracy. Second, in the tests, we compare RF to various comparison approaches, such as statistical, machine learning, and deep learning-based methods, and assess performance using accuracy.

The project can go on by concentrating on increasing the prediction model's precision. To improve the prediction model's accuracy, further advanced prediction models like neural networks and recurrent neural networks may be used. Increasing the prediction model's accuracy can help us advance in proactive failure management. This research work may be further developed by conducting further study on the subject of estimating downtime using prediction analysis.

REFERENCES

[1] M. Sedaghat, E. Wadbro, J. Wilkes, S. D. Luna, O. Seleznjev and E. Elmroth, "DieHard: Reliable Scheduling to Survive Correlated Failures in Cloud Data Centers," 2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Cartagena, Colombia, 2016, pp. 52-59, doi: 10.1109/CCGrid.2016.11.

[2] T. Chalermarrewong, T. Achalakul and S. C. W. See, "Failure Prediction of Data Centers Using Time Series and Fault Tree Analysis," 2012 IEEE 18th International Conference on Parallel and Distributed Systems, Singapore, 2012, pp. 794-799, doi: 10.1109/ICPADS.2012.129.

[3] H. Wang, H. Shen and Z. Li, "Approaches for

Resilience against Cascading Failures in Cloud Datacenters," 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2018, pp. 706-717, doi: 10.1109/ICDCS.2018.00074.

[4] H. Wang and H. Shen, "Proactive Incast Congestion Control in a Datacenter Serving Web Applications," IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, USA, 2018, pp. 19-27, doi: 10.1109/INFOCOM.2018.8485989.

[5] J. Gao, H. Wang and H. Shen, "Task Failure Prediction in Cloud Data Centers Using Deep Learning," in IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1411-1422, 1 May-June 2022, doi: 10.1109/TSC.2020.2993728.

[6] A. Marahatta, Q. Xin, C. Chi, F. Zhang and Z. Liu, "PEFS: AI-Driven Prediction Based Energy-Aware Fault-Tolerant Scheduling Scheme for Cloud Data Center," in IEEE Transactions on Sustainable Computing, vol. 6, no. 4, pp. 655-666, 1 Oct.-Dec. 2021, doi: 10.1109/TSUSC.2020.3015559.

[7] J. Shetty, R. Sajjan and S. G., "Task Resource Usage Analysis and Failure Prediction in Cloud," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2019, pp. 342-348, doi: 10.1109/CONFLUENCE.2019.8776612.

[8] J. Domingos, "Failure Prediction for Cloud Applications through Ensemble Learning," 2021 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Wuhan, China, 2021, pp. 319-322, doi: 10.1109/ISSREW53611.2021.00095.

[9] M. Jassas and Q. H. Mahmoud, "Failure Analysis and Characterization of Scheduling Jobs in Google Cluster Trace," IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society, Washington, DC, USA, 2018, pp. 3102-3107, doi: 10.1109/IECON.2018.8592822.

[10] Y. Xie, D. Feng, F. Wang, X. Zhang, J. Han and X. Tang, "OME: An Optimized Modeling Engine for Disk Failure Prediction in Heterogeneous Datacenter," 2018 IEEE 36th International Conference on Computer Design (ICCD), Orlando, FL, USA, 2018, pp. 561-564, doi: 10.1109/ICCD.2018.00089.

[11] Z. Li, L. Liu and D. Kong, "Virtual Machine Failure Prediction Method Based on AdaBoost-Hidden Markov Model," 2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 2019, pp. 700-703, doi: 10.1109/ICITBS.2019.00173.

[12] A. Marahatta, C. Chi, F. Zhang and Z. Liu, "Energy-aware Fault-tolerant Scheduling Scheme based on Intelligent Prediction Model for Cloud Data Center," 2018 Ninth International Green and Sustainable Computing Conference (IGSC), Pittsburgh, PA, USA, 2018, pp. 1-8, doi: 10.1109/IGCC.2018.8752123.

[13] K. Vani and S. Sujatha, "A Machine Learning Framework for Job Failure Prediction in Cloud using Hyper Parameter Tuned MLP," 2022 Second International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), Bangalore, India, 2022, pp. 1-6, doi: 10.1109/ICATIECE56365.2022.10047809.

[14] M. Soualhia, F. Khomh and S. Tahar, "Predicting Scheduling Failures in the Cloud: A Case Study with Google Clusters and Hadoop on Amazon EMR," 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, New York, NY, USA, 2015, pp. 58-65, doi: 10.1109/HPCC- CSS-ICESS.2015.170.

[15] X. Chen, C. -D. Lu and K. Pattabiraman, "Failure Analysis of Jobs in Compute Clouds: A Google Cluster Case Study," 2014 IEEE 25th International Symposium on Software Reliability Engineering, Naples, Italy, 2014, pp. 167-177, doi: 10.1109/ISSRE.2014.34.

[16] Y. Watanabe, H. Otsuka and Y. Matsumoto, "Failure Prediction for Cloud Datacenter by Hybrid Message Pattern Learning," 2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops, Bali, Indonesia, 2014, pp. 425-432, doi: 10.1109/UIC-ATC-ScalCom.2014.6.