OPEN ACCESS

International Journal of Innovations in Science & Technology

IJIST
P ISSN : 2618-1630
E ISSN : 2909-6130

Original Article

Analysis of Job Failure Prediction in a Cloud Environment by Applying Machine Learning Techniques

Faraz Bashir[1], Farrukh Zeeshan Khan[1]

[1]Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan.

***Correspondence:** Faraz Bashir, **faraz_bashir@rocketmail.com**

Citation | Bashir. F, Khan. Z. F, "Analysis of Job Failure Prediction in a Cloud Environment by Applying Machine Learning Techniques," Int. J. Innov. Sci. Technol., Special Issue, pp. 148-156, 2022.

DOI | https://doi.org/10.33411/IJIST/2022040407

Received | July 01, 2022; **Revised** | Oct 20, 2022; **Accepted** | Oct 25, 2022; **Published** | Oct 30, 2022

Cloud services are the on-demand availability of resources like storage, data, and computing power. Nowadays, cloud computing and storage systems are continuing to expand; there is an imperative requirement for CSPs (Cloud Service providers) to ensure a reliable and consistent supply of resources to users and businesses in case of any failure. Consequently, large cloud service providers are concentrating on mitigating any losses in a cloud system environment. In this research, we examined the bit brains dataset for job failure prediction, which keeps traces of 3 years of cloud system VMs. The dataset contains data about the resources used in a cloud environment. We proposed the performance of two machine learning algorithms: Logistic-Regression and KNN. The performance of these ML algorithms has been assessed using cross-validation. KNN and Logistic Regression give optimal results with an accuracy of 99% and 95%. Our research shows that using KNN and Logistic Regression increases the detection accuracy of job failures and will relieve cloud-service providers from diminishing future losses in cloud resources. Thus, we believe our approach is feasible and can be transformed to apply in an existing cloud environment.

Keywords: Cloud Service Providers, Virtual Machines, Physical Machines, Machine Learning, Infrastructure as a Service.

Acknowledgment.
I am grateful to my research supervisor Dr. Farrukh Zeeshan Khan, for their active support in the timely completion of this research work.

Conflict of interest:
The author had no conflict of interest in publishing this manuscript in IJIST.

Author's Contribution.
All the Author's contributed equally.

Project details.
Nil

Introduction

The usage of cloud computing services is increasing day by day. Cloud consumers expect service providers to supply cloud resources for various ICT services, including business-critical operations, high computing power, scientific computing, and social networking[1]. Because of the huge cloud data centres, resource outages are expected and unavoidable [2]. As a result, ensuring that cloud resources are highly available and reliable is crucial. There is a critical requirement to provide expandable, dependable, and on-demand resources to their users and clients in the event of a defect or failure. Although failures in any computing resource are frequent, massive cloud data centres are set up in such a way as to ensure a certain level of accessibility. IaaS provides computational resources like computing, power, CPU, and memory that ensure high availability. Cloud data centres have an enormous workload. After all, their data are percenters are distributed worldwide [3]. If failures are not handled properly, then the availability of such systems is in danger.

The cloud systems or data centres must be planned in such a way that they face a minimum number of outages [4]. Duplication and backup of data or resources is one solution through which we ensure the reliability and accessibility of cloud resources [5]. Predictive preservation is all about predicting failures and taking action against them. We take data and extract insights from data, and for those insights, we provide future directions based on our observations. Machine learning techniques are appropriate for this purpose since they generate predicted insights from the data collected by these data centres. If we stop all these disasters before they happen, we protect data centres from huge losses. Due to the large amount of data generated by data centres, it is possible to predict when a module is expected to fail or not with the help of ML models[6].

Machine learning's primary purpose is to analyse data structure and provide it into a model that people can understand and use. ML techniques permit machines to study data inputs and generate output values that are inside a specific range using statistical analysis. Thus, Machine Learning (ML) aids computers in developing replicas from selected data to train supervisory procedures based on data inputs. Machine learning is a rapidly growing field. There are two machine learning methods; the first one is supervised learning, which educates algorithms using sample input and output data that individuals have labelled, and the second one is unsupervised learning, which gives the algorithm no labelled data and allows it to find structure in its input data[7].

However, some of the key efforts of scholars in both the academic world and corporate industry predicting cloud-resource failure is still a prime issue in the cloud environment. One of the biggest concerns is assuring and sustaining the accessibility of the whole cloud infrastructure. This is critical because failing to have previous information about a cloud-based failure can have a wide range of consequences, such as failure of any computer-hardware module within any cloud infrastructure resulting in temporary data unavailability. Still, it can also result in permanent data loss in some threatening cases. Furthermore, market forces and new technology trends may coalesce in the future to cause computer hardware system failures to occur more frequently. On the other hand, there are plenty of recommended conventional failure prediction models for dealing with and minimizing the impacts of failures in a cloud environment, but there is a perilous obligation to accurately detect future resource failure patterns. This will not only help in the analysis of future cloud resource failures by modifying existing approaches but also in the planning and development of new methodologies

The main purpose of our research work is to build a precise ML model to predict job failures efficiently in a cloud environment. As a result, we will be able to improve the cloud's reliability and availability by identifying precisely future failures and fully harnessing the potential of next-generation huge cloud computing systems[8].

Literature Review

In [9], [10], the authors proposed a model using time series and ML for failure prediction. They evaluate in their research work that the Support-Vector Machine (SVM) gives the best accuracy among other prediction models. Still, their predicted accuracy can be improved by applying modeltuning to accomplish optimal accuracy for predicting the failures in a cloud system. In [11], the author studied the influence of features linked to the accessibility of distributed storage systems for the google cluster dataset. Their study shows that disk failure can affect eternal data loss, but a significant failure in the cloud data center is a transitory node failure. They developed a model based on Markov chains which describe historical and upcoming accessibility of resources. In [12], the author proposes a new algorithm called HORA. This algorithm predicts both software and hardware failures based on Bayesian networks. However, such learning requires a large amount of data with Bayesian networks. Such knowledge requires a large amount of data with Bayesian networks.

While in [13], the authors proposed a technique called HPC logs which uses LSTM for efficient prediction, and the three-phase Deep Learning technique is used. Firstly, logs are trained, and then chain recognition of events is also introduced, and during the last stage, lead time is predicted during the test, but on the other hand, LSTM takes a huge time to prepare and requires additional memory to train. In [14] RNN model is proposed for the failure prediction of hard drives. The SMART dataset is used in the study. RNN cannot handle long-term dependency, so that's why it is not suitable for these kinds of predictions. The authors used a collective classifier in [15] to predict hard drive failure in a cloud environment. They use data from two sources to perform their research study. They used to concentrate on failures of hard disks in the cloud environment in their studies, but business infrastructure relies on other modules, not only a hard disk.

In [16], the author studied the failure analysis of VM and PM, which are hosted commercial data centres. The author found out that the failure patterns of both these are different. The failure of VM is lower than that of PM. By increasing the computational complexity of VMs, their failure rate cannot improve. LSTM for job failure prediction is also used in [17][18]. In [19] authors proposed failure based on SVM and Random Forest. RF is used for a sequence of operations, and SVM is used for classification purposes in the study. While in [20], the author proposes a forecasting approach based on GARCH and ARIMA, which predicts the time between failures and response time. In this study, various features like memory, Disk, I/O time to organize failed disks, and good disks are extracted. HSMM, in general, cannot lever a classification of data or very high-dimensional data.

Authors in [21] proposed a unique algorithm and predicted failures in the hard drives called (MI-NB) which uses Naïve Bayes as a classifier. The author compares the performance of his model with SVM, but SVM is computationally expensive. While in [22], [23], the author proposes Arima mode and fault tree analysis for prediction purposes. The framework alerts the cluster resource if a failure is going to happen in the computing environment, and appropriate actions are considered.

However, plenty of proposed cloud failure prediction methods for dealing with and diminishing the consequences of failures in a cloud environment. There is a critical need to efficiently recognize one of the finest models for future job failure patterns. Thus, we performed comparison-based testing for higher prediction accuracy by applying machine learning algorithms to evaluate the outcome for optimal accuracy and effective results.

Methodology

There are several steps in the machine learning process. Firstly, the data preprocessing step removes noise from the dataset, and all the outliers are removed. To build a reliable and fault-tolerant system, it is necessary to do preprocessing on collected data. Secondly, a feature selection technique is applied in which we select features based on the feature selection technique. Optimal feature selection is necessary for a machine learning model to perform well, and a chi-square technique is used to select relevant features in our model. We aim to pick features that are highly responsive to the response. When two features are independent, the perceived count is close to the predicted count, and the chi-square value is smaller.

On the other hand, a high chi-square value specifies that the hypothesis of individuality is incorrect. Therefore, we don't need all the features for our model. It may overfit or underfit our model. Thirdly, we build our model to train for job failure predictions. Lastly, we have a prediction stage where all the prediction takes place. Therefore, we use two machine learning algorithms for job failure prediction discussed below in detail. Figure 1. shows the overview of the process evaluation used for our ML models.

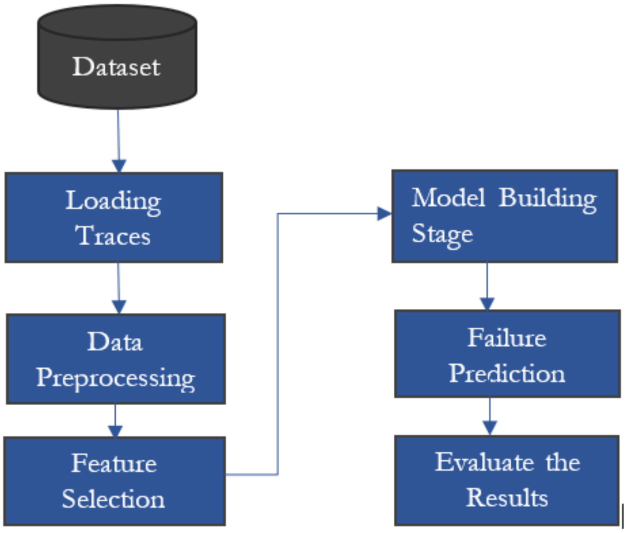

Figure 1. Overview of process evaluation

In our process evaluation, we build our dataset by pre-processing and then load the traces by selecting relevant features that help in Machine Learning Model. We developed a precise Machine Learning model for predicting cloud job failures in the model-building stage. Lastly, our proposed work compared two different Machine Learning Algorithms that apply to the dataset and then concluded which Algorithm is the most accurate and found the best classifier or Machine Learning Algorithm in which prediction accuracy increased as compared to the literature view.

Dataset Overview

For experimentation purposes, we use the Bit brains dataset, which contains traces of 1750 VMs. Bit brains have features like Timestamps, CPU cores, CPU capacity, memory usage, and Disk read throughput. The data is collected in two-time intervals, one in 2017 and the second one in 2019. In our dataset, we have one target class holding each job's status. Like 0 for failure, 1 for completion, and 5 for a partially completed job.

Data Pre-processing

The first step is data preprocessing. In this step, noise is removed from the dataset, and all the outliers are removed. To build a reliable and fault-tolerant system, it is necessary to do processing on collected data. Collected data is not clean and contains outliers, so the preprocessing step is mandatory in data preparation. Our collected data contains additional pointless information and corrupt data. We need only usable and consistent data that is helpful in our model for predicting failures. After cleaning, traces of 1378 VMs are used in our model. Our prediction model uses additional features for prediction performance:

1. CPU Cores are individual processing units in a computer system. Today our computers have multiple cores. It is evident from the research that the job or task which contains more CPU cores is less likely to fail.
2. CPU Usage is another important aspect of task or job failure. The job with higher CPU usage is more likely to kill or fail.
3. Scheduling delay is waiting time for each task. It is found that the jobs that cannot finish their task have higher scheduling delays.
4. Task Priority is a priority given to a task based on its completion time. The jobs with the highest and lowest priority are more likely to fail than those with middle task priority.

We preprocess our data and obtain features that are optimal for our experiment.

Logistic Regression

Logistic regression is used for both classification and regression. It is one of the vital algorithms in machine learning. Whenever we have a relationship between two variables, then we use regression algorithms. In classification problems, it is mainly used for binary classification problems.

Our model uses logistic regression in the job failure prediction method. We specifically used logistic regression because our target variable is categorical. In our model status of the Disk are the label and target class. Label class contains three target sets, namely 0, 1, and 5. 0 means our job is failed, means completed its task, and 5 means the job terminates in the middle. The label class is dependent, and it depends upon CPU memory, CPU usage, and CPU cores. These are the features that are selected through the feature selection technique. Logistic regression takes the independent variable at the x-axis and the dependent variable at the y-axis. In logistic regression sigmoid function is used for classification. The sigmoid function puts the value of the dataset in the ranges 0 and 1. In the sigmoid function, when the value of the target variable is negative, it assigns a 0 value, and if the value is infinity, then it assigns 1. In this way, we obtain a value between 0 and 1.

K-Nearest Neighbour (KNN)

KNN is the most important and most used algorithm used in machine learning. KNN works based on distance. It calculates the distance between the points. Euclidean distance is mostly used in KNN. It is used for classification and regression. KNN assumes that similar things are present nearby. The things which have similar characteristics are placed closer to each other.

In our model, we use KNN to compute the distance between features; features with similar distances are placed in the same cluster. In KNN, the value of K is computed using the elbow method. We take the k value as 3 in our model; an optimal value of k is necessary to make our model appropriate for classification.

Result and discussion

Individual metrics like precision, recall, and F-measure is used to assess the machine learning model's performance regarding accuracy. The confusion matrix is used to calculate the values of these metrics. We calculate values like true-positive, true-negative, false-positive, and false-negative. Based on these values, we conclude whether our ML model is performing well or not. A good machine learning model should achieve high true and low false positives.

Table 1. shows the complete Evaluation Metric, where accuracy is the percentage of all the correctly identified instances and records in the dataset. Precision is all the positive predicted values of the dataset. Recall tells us about the number of positive predicted values among all the positive predictions. F-measure is the measure of test accuracy.

Table 1. Evaluation metric

Evaluation Metric	Equation
Accuracy	(TP + FP) / (FP + TP + FN + TN)
Precision	TP / (FP + TP)
Recall	TP / (TP + FN)
F-measure	(2*P*R) / (P + R)

We differentiate our ML model from other models. Using the test set, which contains three kinds of job failures. Different metrics were computed using the confusion matrix. The accuracy achieved by our model logistic regression and KNN is 95% and 99%. It is evident from the results that these two algorithms perform well in predicting failures in the cloud infrastructure. Logistic regression expresses the relationship between variables of the datasets, and KNN places variables near each other based on distance or similarity. Logistic regression and KNN achieve higher accuracy, precision, recall, and f-score. Table 2. illustrates the performance evaluation metric of the Logistic-Regression algorithm.

Table 2. Classification Report of the Logistic Regression Algorithm

Classes	Precision	Recall	F1-score
0	0.81	0.11	0.20
1	0.97	0.97	0.97
5	0.85	0.90	0.88
Accuracy			0.95

It is evident from the classification report that the accuracy achieved by logistic regression is 95%. In the dataset, 1 means the job is completed, 5 means the job terminates in the middle, and 0 means the job failed. 85% precision is achieved on 5 and 97% on 1. Table 3. shows the performance evaluation metric of the KNN algorithm.

Table 3. Classification Report of KNN Algorithm

Classes	Precision	Recall	F1-score
0	0.92	0.83	0.87
1	0.99	1.0	0.99
5	0.98	0.96	0.97
Accuracy			0.99

It is evident from the classification report that the KNN achieves an accuracy of 99% with very high precision in both classes 1 and 5. It predicts accurately.

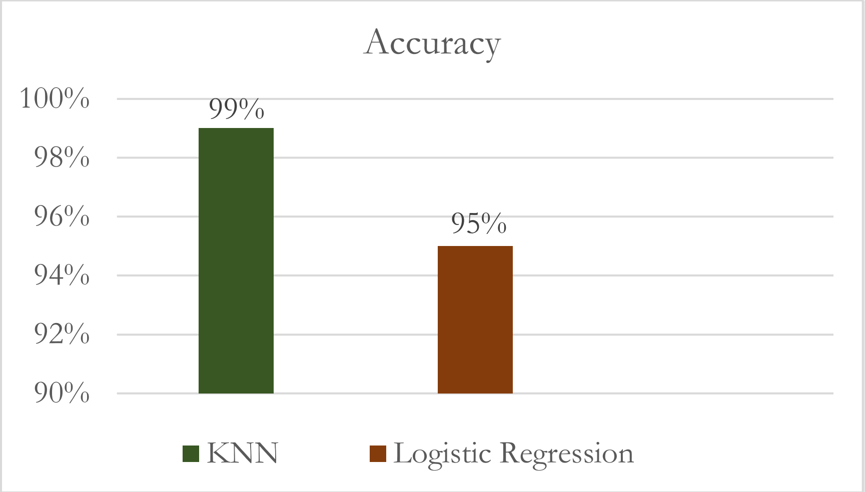

Figure 2. Accuracy of KNN and logistic regression algorithm

Figure 2. shows an accuracy graph of all two algorithms which are evaluated, including accuracy, precision, recall, and f-measure. KNN and Logistic-Regression perform well in predicting job failures.

Figure 3. shows the complete performance evaluation metric of KNN and the Logistic-Regression algorithm. It is evident from the graph that the precision, recall, and f-score of KNN and Logistic Regression are high. Therefore, we conclude from our results that the KNN and Logistic Regression is a suitable algorithms for job failure prediction.

The studies related to failure prediction in a cloud environment using machine learning are shown in Table 4. When a table is examined, it is determined that the different machine learning algorithms are used for this purpose. Various datasets are used for this purpose. The use of these datasets is positive for comparing the performance of machine learning algorithms with our research work.

However, it is indeed that timely and effective fault prediction is perilous. We need to comprehend the reasons behind any failure. Some reasons for cloud failure are human mistakes, cloud-provider downtime, severe spikes in client requests, third-party facility failures, and storage failures. These types of failures may also lead to massive economic losses in a huge cloud system, but these losses can be avoided by dealing with the failures confronted by the cloud systems in a real-time environment.

Our study shows that machine learning models successfully predict failures in a cloud environment. Our work aims to contribute to the research community. The outcome of our research work is to examine the importance of analyzing and accurately predicting cloud job failures and develop an accurate ML model for failure prediction that gives optimal results in predicting cloud job failures. We evaluate the results by comparing them with the previous related work. Our experimental results show that using KNN and Logistic Regression increases the detection accuracy of job failures by using the feature selection technique. Features are selected verily based on feature selection techniques. It can be said that the machine learning model classification contributes positively to the classification of failure detection when used with feature selection methods. However, the feature selection method does not always give the optimum number of features.

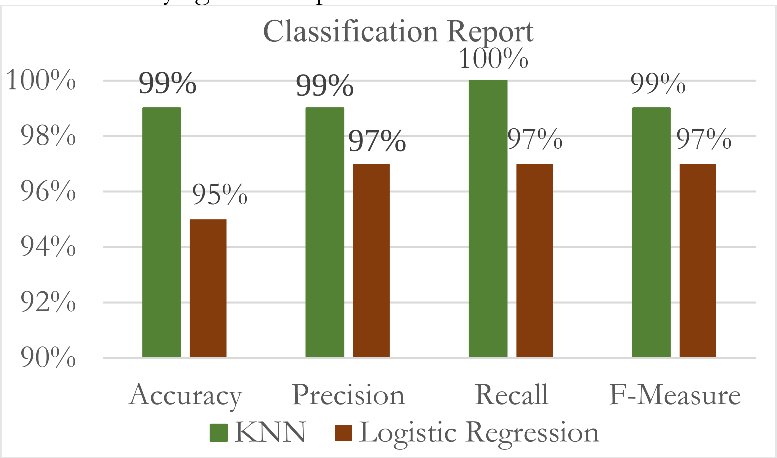

Figure 3. Classification report of KNN and logistic regression algorithms

TABLE 4. PERFORMANCE COMPARISON WITH PREVIOUS RELATED WORK

Research Papers	Effective Machine Learning Algorithm	Accuracy
Task Failure Prediction in Cloud Data Centres Using Deep Learning [6]	Bi-LSTM	87%
Failure prediction using machine learning in a virtualized HPC system and application [7]	Support Vector Machine (SVM)	90%
Task Resource Usage Analysis and Failure Prediction in Cloud [8]	XGBoost	92%
Predicting Application Failure in Cloud: A Machine Learning Approach [13]	RNN named LSTM	87%
Failure Analysis Modelling in an Infrastructure as a Service (Iaas) Environment [12]	Auto-Regressive-Moving Average (ARMA)	95%
Proposed Methodology	KNN, Logistic Regression	99% , 95%

Conclusion

Developing new strategies for predicting job failures in a cloud environment is an agile and demanding problem. Some of the previously proposed methods are based on machine learning techniques, able to adjust to specific circumstances but are unsuccessful in various environments. The failure prediction dataset is obtained from the bit brains in our research study. Bit brains dataset contains data about the resources used in a cloud environment. The dataset includes statistical features like average waiting time and CPU cores. The Chi-square technique is used to select relevant features from the dataset. More than 100 thousand records are selected for training purposes. Our experimental results show that KNN and Logistic Regression give optimal results with an accuracy of 99% and 95%, respectively Our results show that our machine learning model successfully predicts job failures in a cloud environment and gives the highest prediction accuracy and precision-recall rate as compared to previous related work. Therefore, we conclude from our results that the KNN and Logistic Regression is a suitable algorithms for job failure prediction in a cloud infrastructure. In the future, we intend to examine enormous publicly accessible cloud datasets by applying multiple Machine Learning strategies or techniques and comparing them to get a more precise prediction accuracy.

Reference

[1] M. S. Jassas and Q. H. Mahmoud, "Analysis of Job Failure and Prediction Model for Cloud Computing Using Machine Learning," Sensors 2022, Vol. 22, Page 2035, vol. 22, no. 5, p. 2035, Mar. 2022, doi: 10.3390/S22052035.

[2] M. S. Ajmal, Z. Iqbal, F. Z. Khan, M. Ahmad, I. Ahmad, and B. B. Gupta, "Hybrid ant genetic algorithm for efficient task scheduling in cloud data centres," Comput. Electr. Eng., vol. 95, p. 107419, Oct. 2021, doi: 10.1016/J.COMPELECENG.2021.107419.

[3] J. Gao, H. Wang, and H. Shen, "Task Failure Prediction in Cloud Data Centres Using Deep Learning," IEEE Trans. Serv. Comput., 2020, doi: 10.1109/TSC.2020.2993728.

[4] D. Cotroneo, L. De Simone, P. Liguori, and R. Natella, "Enhancing the Analysis of Software Failures in Cloud Computing Systems with Deep Learning," J. Syst. Softw., vol. 181, Jun. 2021, doi: 10.1016/j.jss.2021.111043.

[5] B. Mohammed, I. Awan, H. Ugail, and M. Younas, "Failure prediction using machine learning in a virtualised HPC system and application," Clust. Comput. 2019 222, vol. 22, no. 2, pp. 471–485, Mar. 2019, doi: 10.1007/S10586-019-02917-1.

[6] J. Shetty, R. Sajjan, and G. Shobha, "Task resource usage analysis and failure prediction in cloud," Proc. 9th Int. Conf. Cloud Comput. Data Sci. Eng. Conflu. 2019, pp. 342–348, Jan. 2019, doi: 10.1109/CONFLUENCE.2019.8776612.

[7] B. Mohammed, B. Modu, K. M. Maiyama, H. Ugail, I. Awan, and M. Kiran, "Failure Analysis Modelling in an Infrastructure as a Service (Iaas) Environment," Electron. Notes Theor. Comput. Sci., vol. 340, no. October, pp. 41–54, 2018, doi: 10.1016/j.entcs.2018.09.004.

[8] S. W. and I. M. Sehir e N, Shehzad M.A, Aslam M.S, "Optimize Elasticity in Cloud Computing using Container Based Virtualization," Int. J. Innov. Sci. Technol., vol. 2, no. 1, pp. 1–16, 2020.

[9] T. Islam and D. Manivannan, "Predicting Application Failure in Cloud: A Machine Learning Approach," Proc. - 2017 IEEE 1st Int. Conf. Cogn. Comput. ICCC 2017, pp. 24–31, Sep. 2017, doi: 10.1109/IEEE.ICCC.2017.11.

[10] C. Liu, J. Han, Y. Shang, C. Liu, B. Cheng, and J. Chen, "Predicting of Job Failure in Compute Cloud Based on Online Extreme Learning Machine: A Comparative Study," IEEE Access, vol. 5, pp. 9359–9368, 2017, doi: 10.1109/ACCESS.2017.2706740.

[11] D. Ford et al., "Availability in globally distributed storage systems," Proc. 9th USENIX Symp. Oper. Syst. Des. Implementation, OSDI 2010, pp. 61–74, 2019.

[12] T. Pitakrat, D. Okanović, A. van Hoorn, and L. Grunske, "Hora: Architecture-aware online failure prediction," J. Syst. Softw., vol. 137, pp. 669–685, Mar. 2018, doi: 10.1016/J.JSS.2017.02.041.

[13] A. Das, F. Mueller, C. Siegel, and A. Vishnu, "Desh: Deep learning for system health prediction of lead times to failure in HPC," HPDC 2018 - Proc. 2018 Int. Symp. High-Performance Parallel Distrib. Comput., pp. 40–51, 2018, doi: 10.1145/3208040.3208051.

[14] C. Xu, G. Wang, X. Liu, D. Guo, and T. Y. Liu, "Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks," IEEE Trans. Comput., vol. 65, no. 11, pp. 3502–3508, Nov. 2016, doi: 10.1109/TC.2016.2538237.

[15] S. Ganguly, A. Consul, A. Khan, B. Bussone, J. Richards, and A. Miguel, "A Practical Approach to Hard Disk Failure Prediction in Cloud Platforms: Big Data Model for Failure Management in Datacentres," Proc. - 2016 IEEE 2nd Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2016, pp. 105–116, May 2016, doi: 10.1109/BIGDATASERVICE.2016.10.

[16] R. Birke, I. Giurgiu, L. Y. Chen, D. Wiesmann, and T. Engbersen, "Failure analysis of virtual and physical machines: Patterns, causes and characteristics," Proc. Int. Conf. Dependable Syst. Networks, pp. 1–12, Sep. 2014, doi: 10.1109/DSN.2014.18.

[17] X. Chen, C. Da Lu, and K. Pattabiraman, "Failure prediction of jobs in compute clouds: A Google cluster case study," Proc. - IEEE 25th Int. Symp. Softw. Reliab. Eng. Work. ISSREW 2014, pp. 341–346, Dec. 2014, doi: 10.1109/ISSREW.2014.105.

[18] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," Proc. ACM Conf. Comput. Commun. Secur., pp. 1285–1298, 2017, doi: 10.1145/3133956.3134015.

[19] I. Fronza, A. Sillitti, G. Succi, M. Terho, and J. Vlasenko, "Failure prediction based on log files using Random Indexing and Support Vector Machines," J. Syst. Softw., vol. 86, no. 1, pp. 2–11, Jan. 2013, doi: 10.1016/J.JSS.2012.06.025.

[20] A. Amin, A. Colman, and L. Grunske, "An approach to forecasting QoS attributes of web services based on ARIMA and GARCH models," Proc. - 2012 IEEE 19th Int. Conf. Web Serv. ICWS 2012, pp. 74–81, 2012, doi: 10.1109/ICWS.2012.37.

[21] J. F. Murray, G. F. Hughes, and K. Kreutz-Delgado, "Machine learning methods for predicting failures in hard drives: A multiple-instance application," J. Mach. Learn. Res., vol. 6, no. May 2014, 2005.

[22] T. Chalermarrewong, T. Achalakul, and S. C. W. See, "Failure prediction of data centres using time series and Fault Tree Analysis," Proc. Int. Conf. Parallel Distrib. Syst. - ICPADS, pp. 794–799, 2012, doi: 10.1109/ICPADS.2012.129.

[23] Q. Guan, Z. Zhang, and S. Fu, "Ensemble of Bayesian predictors and decision trees for proactive failure management in cloud computing systems," J. Commun., vol. 7, no. 1, pp. 52–61, 2012, doi: 10.4304/jcm.7.1.52-61.