Cloud Data Centers Based Task Failure Prediction Using Machine Learning Techniques

Regalla Phaneendra[a]*, Arif Mohammad Abdul[b]

[a]*Regalla Phaneendra,Hyderabad, India*
[b]*Arif Mohammad Abdul, Hyderabad, India*

Abstract

The highly scalable cloud data centre should dispense efficient consistency as well as scalability with a low failure probability. Meanwhile, large scale cloud data centres tend to have significant failure rates due to a variety of factors, including software and hardware failure, that also has higher probability in task failures. Severe mishaps will drastically impair overall dependability of cloud services whilst still consuming a lot of resources for revive the services out of failure. To prevent unanticipated loss, this is essential for predict the task failure having great accuracy earlier they arise. In this study, consequently provide an exhaustive comparing as well as evaluation metrics using prediction methods for task failure. Predominant algorithms of machine learning are employed to develop and test these techniques. During testing and training these classifiers, used a reference data-set called Google Cluster data. The research yielded the following results. We concluded that Random Forest yields the most accurate model for predicting task failure with accuracy score of 94.625%, Precision and F1 scores with 0.89, 0.74. Decision tree achieved 0.69 of Recall score.

Keywords: Cloud; Precdicting; Machine laerning; Prediction; Task failure.

Nomenclature

PM	Physical machine
VM	Virtual machine
CPU	Central processing unit
OS-SVM	Online sequential-support vector machine
SVM	Support vector machine
RF	Random Forest
DT	Decision Tree
LR	Logistic egression
KNN	K-nearest neighbour
VC	Voting classifier
NB	Naïve bayes

1. Introduction

Cloud services entails the delivery of numerous services which including storage of data, infrastructure, networks and applications over the internet. Such method is gaining popularity among clients that require storage space and businesses looking for a secure website backup data service. Pay as you go should suit user's cloud global resources, thus they don't have to fret about overprovisioning a business which consumption of resources doesn't quite actually deliver, and instead squandering money under-providing the services that becomes important in the market, and thereafter foregoing out on additional income.

Providers of cloud services use server virtual machines that let a physical server (PM) to perform several virtual machine instances (VMs) jobs using distinct utilization of resources. The VMs in a cloud support legion application. Since the workload upon the PM fluctuates by schedule, the PM could be overworked. That traffic unbalance during a PM has a negative impact on the performance of every virtual application executing upon a PM. Since cloud - based data services are frequently overloaded, services like CPU and internet were associated with significant morbidity due to being shared by multiple users. Fig.1. illustrates overall architecture of existing data center of cloud. Users can transmit requests to a cloud, which including data storage and performing applications. Every cloud data center has been comprised of physical machine (PM), which can each handle the cluster of virtual machines (VM). Tasks sent through individuals were executed within every virtual machine. A large-scale data center may host a large

* Regalla Phaneendra
E-mail address: phaniram4@gmail.com

number of machines, each of which execute numerous apps and get client requests from users around the world each millisecond. The distributed storage facility of such variability as well as intense loads could be exposed to several forms of failure at times.

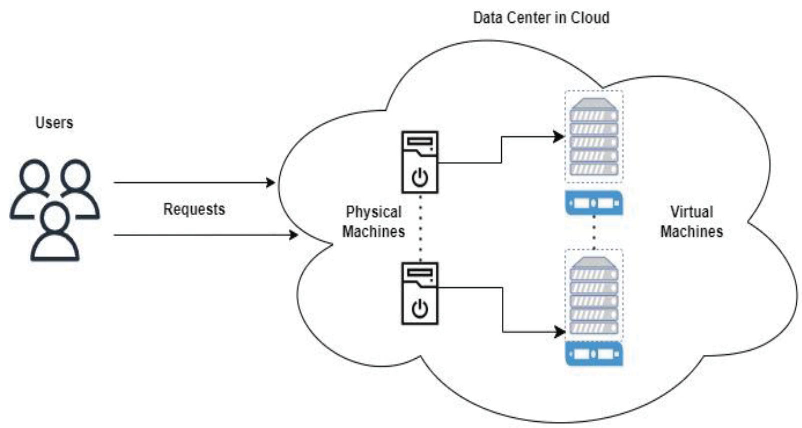

Fig. 1. Illustration of cloud data center

Earlier research [1], [2] has shown revealed failures of hardware and disc, seems to be a main cause of cloud service disruptions. But since failure of disc might resulting not only a service disruption and moreover immutable loss of data. Whilst each failure of disk is uncommon, the device containing several many disks can frequently encounter failure. Therefore, effective prognosis of application failure ahead of time will enhance the effectiveness in overcoming such fault while keeping the application active.

2. Related work

The above part provides a comprehensive literature review, along with a study as well as overview on prognosis of failure. Majority of the studies have seen a significant increased interest in failure analysis applying machine learning approaches for a large and complex excellent efficient in environment of cloud. To thoroughly ascertain and anticipate higher efficiency cloud environment metadata practically employing the failure on actual data, only several studies have been conducted.

The study [3] compare the effectiveness of relevant state-of-the-art prediction of workload algorithms in this research. Proposed an approach for performing such prognosis ahead of time in so that there is enough period to scheduling tasks forecasting workloads. For increase predictive performance even further, we propose a clustering-based prediction of workload strategy that splits overall tasks onto various factions before training the predicting approach for every set. The Google cluster trace-driven results demonstrate clustering-based workload estimation techniques outperforms previous comparable approaches but also achieved better predictive accuracy across over 90% through both memory and CPU. [4] presents a new two-phase machine learning technique to calculate workflow task completion timeframes with in cloud for variable data input. This approach relies on variables expressing duration statistics with two steps for forecast better accuracy. It obtains best and worst phase outcome of 1.6% and 12.2 %, whereas current approaches yielded inaccuracies of more than 20% since much above 75% of an examined task of workflows. This could replicate very least input onto newer clouds with little failures via needing just a finite numeral of steps.

The data centers of cloud should ensure exceptional dependability services and scalability whilst reducing overall possibility of failure. Inadequacies might substantially impact the dependability on services of cloud while also costing a large number of assets to recover the service. For enhance failure predictive performance even further, this study [5] illustrates a predictive model that employs multi-layer Bidirectional Long Short-Term Memory (Bi-LSTM) for detect task and job failure with in cloud. The Bi-LSTM algorithm's intent is to estimate whether tasks and jobs will fail or succeed. Our approach surpasses established algorithms, achieving 93% accurate result for task failure and 87% accuracy rate for job failures. Earlier activity breakdown estimation and appropriate recycling processes may considerably enhance asset capacity in high scaling data centers. For addressing this issue, work [6] proposes a unique approach rely on online sequential extreme learning machine (OS-ELM) to evaluate online job status of cessation. A comparison analysis employing google trace data reveals that its hypothesized technique's accurate rate is 93%, with the prototype upgrading within 0.01 seconds. As compared to approaches like support vector machine (SVM), ELM and OS-SVM, this technique offers several benefits, including less time generating and upgrading the system, greater predictive accuracy and precision, and good efficiency of false negative. Applications of Scientifical use the cloud to implement specific processes using tasks. Whenever task fails, the reliance orientation including the workflow takes into entire execution efficiency. This study [7] presents the predictive failure strategy that integrates a combination of machine learning techniques. For better accuracy, ensembles naive Bayes with various classifiers and determine here that the naive Bayes combined random forest as well as MF2N2 classifier attains accurate results (95.8%) compared to any other ensemble technique. This suggested approach has verified through combining simulated as well as actual cloud infrastructure.

Under this research [8], primarily investigated and classified the activity of failures as well as performed tasks via accessible to the public data. Conceived and built a failure predicting technique to detect cancelled tasks before they happen. This proposed method attempts to improve machination utilization and even the efficiency of cloud. By analyzing the conceptual approach using three openly accessible records like Mustang, Trinity and goggle cluster. Furthermore, data records being submitted to several machine learning algorithms in order to estimate the more precise option. Ultimately, by utilizing Random Forest and Decision tree classifiers attained better accurate results for Google trace data with parametric metrics like recall, precision and F1 score. Failure of cloud services is a significant concern because it could cost facility donors of cloud huge amount of billings, particularly alongside the dissipation of productivity incurred to business clients. As a result, in this work [9] a comparative evaluation as well as model validation for estimating approaches for task failures in application. For training and testing the methods, used reference dataset termed google cloud traces. Machine learning and deep learning techniques are applied to build and train these data traces. By the results, determined that in the instance for job failure estimation, Extreme Gradient Boosting gives the best accurate outcome, with features of disc space and CPU request. And for scaled evaluation, Logistic Regression method is the most flexible.

This is critical in a massive data center, that accurately discern application termination tracking. Several supervised learning methods has been employed towards this problem, since it is effective for enhancing resource utilization efficiency. Due to this sometimes the predictive accurate rate will be alleviated. This research [10] proposes a novel prediction failure methodology on the alliance connections among comparable jobs which can collectively estimate task dismissal status updates with an incipient time. Initially, the job cluster technique is developed for selecting tasks with greater resemblance among tasks with varying task counts. Finally, relying upon that task cluster outcomes, its robust multi-task learning approach is proposed in order to optimally use information across tasks. These conclusions reveal the suggested approach achieves greater predictive rate, reduced mis judgement rate. A statistical review of cloud existing workload metadata offers visibility to failure attributes, that can be applied to boost system solidity. This article [11] summarizes detailed systemic review on task utilization statistics on the Google cluster data, followed by the design of failure estimating approaches to foretell failures. This is revealed that its resource utilization behavior, processing period, & volume of resource consumed either by failure task differs from a completed task. Multiple interpolation strategies as well as the XGboost algorithm were used to anticipate the failure of a job in a massively contrast dataset, after it was revealed that Synthesis minority oversampling and the XGboost classifier detected the task report with 94.8% recall and 92% precision.

Understanding and categorizing reported failures is critical for building the secure cloud infrastructure. Purpose of this study [12] is to discover the critical elements which correspond to cloud application faults but also proffer the prediction approach which could forecast the conclusion about a job before it completes, crashes, or get killed. To achieve, conducted a failure classification analysis on Google cluster workload traces. Investigation shows both failed and terminated jobs consume a large number of resources. For investigate the possibilities for failure detection in applications of cloud in order to decrease resource waste by improving the tasks and jobs that inevitably fail or killed. For predict cloud application errors, offers a predictive procedure entrenched upon the special form of Recurrent Neural Network (RNN) called Long Short-Term Memory Network (LSTM). This algorithm predicts task failures having 87% accuracy, has a 85% of true positive rate, as well as false positive rate of 11%. Service suppliers of Cloud infrastructure are accountable to regulating its accessibility for allocating computational operations to furnish their clients with high quality of service. Examined and assess three publicly accessible enormous clustered data via Alibaba, Google and Trinity for characterizing failure of task on platform of cloud computing. By [13] devised the paradigm of failure aware task scheduling, which capable of predicting the terminating state about group of assigned tasks within execution as well as adopting relevant proactive measures. Artificial and convolutional neural networks are used for estimating the activities. By using Google data collection, the outcome demonstrate that ANN and CNN will attain predictive accuracy levels of 94% and 92%, respectively. Furthermore, with employing the appropriate process, the platform can prevent nearly 40% operations expected to fail utilizing Alibaba data, preserving significant ensembled resources like RAM and CPU.

Node failures in platform like AWS could really decrease the accessibility of its cloud infrastructure as well as possibly result in huge loss of revenue. Modeling failures of node may lead to critical because that allows DevOps developers could mitigate the severity through taking pre-emptive measures. Moreover, those prognostications were difficult because of numerous constraints, such as the massive bulk of the monitoring statistics and also the intricacy of a breakdown characteristics. By employing the techniques of machine learning, [14] considering data strategies includes oversampling and gapped. The analysis reveals that using random forest in combination with oversampling approach produces the greatest results in case of computational expense and predictive ability.

3. Dataset Description and Preprocessing

Researchers are already capable of solving computer issues throughout a short amount of time which would require many years upon the computer due to the expansion of computational facilities. Although the likelihood of an application failing due to a software or hardware problems improves as scalability rises. This type of operational error not just to impedes technological discovery and also consumes the significant quantity of time. By the previous research [15], the dataset containing parameters of “jobid”, “memory”, “networklog10”, “localIOlog10”, “NFSIOlog10” and “failed”. As shown in Table 1. The term failed is the output parameter and remaining are the input parameters.

The dataset's columns are as follows:

- **jobid:** the specific reference for every job.
- **memory:** the job's memory consumption, measured in Gigabytes.
- **networklog10:** its connectivity bandwidth utilized by a task in MB per second, log base 10.
- **localIOlog10:** a local Input/Output bandwidth utilized either by task in MB per second, log base 10.
- **NFSIOlog10:** Its value, expressed in MB per second, is the log base 10 of the input/output bandwidth required by the task on through network file system (NFS).
- **failed:** a binary indication that illustrates if the work was successful or unsuccessful, with a value of 1 denoting fail and a value of 0 success.

Table 1. Parameters in dataset

Parameters	Labels/Variables
Input	jobid memory networklog10 localIOlog10 NFSIOlog10
Output	failed

The dataset contains of 20000 job/task log files with the above-mentioned input and output parameters. We have information upon every job's resource utilization as well as the whether or not it was successful. It has failure data around approximately 8%, the positive class is what we refer to it as. “1” refers to failure and “0” refers to success. The replica of dataset is mentioned in Table 2.

Table 2. Dataset catalogue

Sr.No	1	2	…	…	20000
jobid	jobID 1634295	jobID 2033452	…	…	jobID 1165709
memory	44.3904	31.5839	…	…	7.9835
networklog10	-1.0262	-1.4608	…	…	0.7225
localIO10	0.8033	-0.608	…	…	-1.1583
NFSIOlog10	-3	-2.9967	…	…	0.2308
failed	0	0	…	…	0

4. Methodology

An experimental setup as well as the methodological approach enabling failures evaluation and prediction were addressed in this portion. Fig. 2. Illustrates to conduct an extensive comparative on the prediction of task failure based on the dataset. Here, we classified an operation into three segments, notably “Data handling”, “Model building”, “Analysis”. Data handling is the action of retrieving information from available records, which includes task occurrences, and then process the dataset. Segment model building necessitates the actions for building & testing prediction model built using machine learning techniques. This proposed methodology was aimed at solving the classifying issue, i.e., characterize anticipated dismissal outcome of every job, whether it is success or failed. The dataset has split among 80% as training as well as 20% as testing. These algorithms at work operate as follows. Segment analysis can evaluate the efficiency for prediction models using multiple statistical measures for identify the most effective algorithm.

Fig. 3. Illustrates the job event table's dispersion across scheduled groups which displays the status of the task failed/success. In above graphical representation, “1” refers to failure and “0” refers to success.

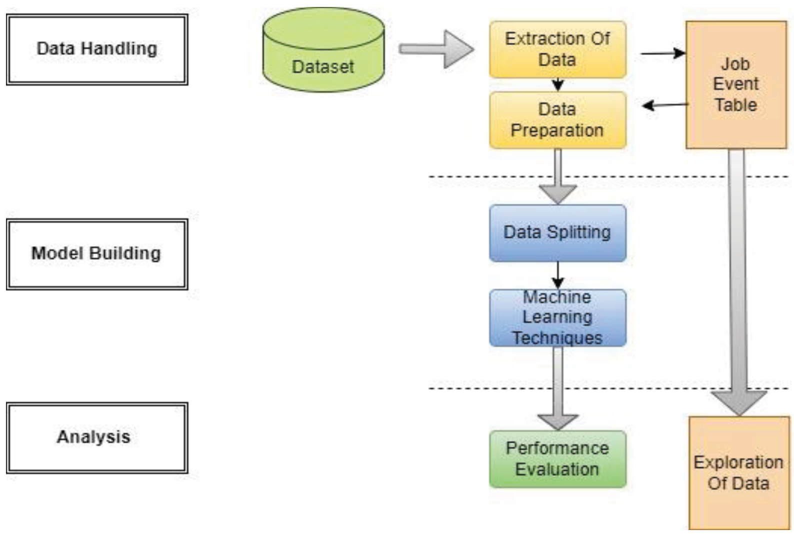

Fig. 2. Architecture illustration

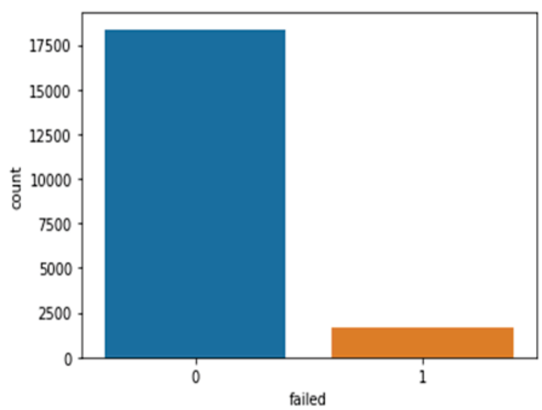

Fig. 3. Distribution of failed status

4.1. Algorithms used

Algorithms based on machine learning are also commonly employed in virtual machine job scheduling. The several machine learning techniques are employed and evaluated. In this study, machine learning techniques like Decision tree, Random Forest, KNN, Support vector machine, Logistic regression, Naïve bayes and Voting classifier are deployed and also used few statistical metrics.

- Support vector machine:
  This is centred upon that computing of such a linear regression model in such a multi – variate classifier at which inputs could be utilised through a non-linear regression model [16]. Since the computational complexity in constructing the model need not include additional data for training that is closest towards the prediction's outcome, each trained model via SVM is only reliant upon the sample from the set of training. From another terms, well during training the SVM classifier, one applies additional weights for the sample points beyond the prior expected to ensure that the method may explore further also on observations identify additional potential trends in the data set records.

- Logistic regression:
  Logistic regression will determine the one or more varying components getting evaluated and seems to have the significant correlation including an output as well as presents the indicator for the extent of future impact [17]. Additionally, it may balance the influencing variables, which are contributing factors with those related predicting parameters and indeed the result, ensuring that the estimate from the impact including the predicting on relevance also isn't affected more by impact.

- Random Forest:
  Random forests were a tree classifier integration whereby every tree is dependent upon that random vector's values, which are finite element modelling with a uniform occurrence overall the forest's trees. As both the number of trees inside a model expands, the generalisation error corresponds toward a limitation. This amplitude for every distinct baseliner as well as the connectivity amongst individuals define a generalisation error among a forest of classification and regression trees. A threshold again for generalisation error in this model can be determined using the combination of two factors which represent the accuracy of the different classifiers and indeed the independence of those models.

- Decision Tree:
  The formalisation of describing these conversions is therefore a decision tree. The leaf node annotated including a classification made up of a node connected to two maybe more sub - trees constitutes a tree. Each potential consequence was connected to a respective sub - trees, but a node evaluates particular interesting take upon that feature labels of an occurrence. Commencing with the tree's root node, an occurrence was categorised. Whether this tree would be a trial, then instance's result gets obtained as well as the procedure then is carried out by employing the suitable sub-tree. Every label of a leaf indicates its anticipated subclass of an occurrence that will finally be examined.

- K-nearest neighbor:
  This model has been a basic and efficient segmentation approach. The algorithm is a supervised learning classification that is non-parametric as well as employs approach for classify and otherwise predict that how given data value will be grouped. For every test sample, k nearest neighbours (KNN) should determine the distance (aka similar) across all training dataset [18].

- Naïve bayes:
  For a classified parameter, this Naïve bayes (NB) classification model is one family using elementary posterior probability that is widely chosen also as reference on classification tasks. This is built around the standard hypothesis that over parameters were dependent from one another [19].

- Voting classifier:
  A proposed feature selection voting classification combines three algorithms (i.e., support vector classification, random forest, decision tree) of machine learning techniques which integrate classifier well with ensembles method of voting [20]. This is a method which builds from an ensemble containing different approaches but also estimates the outcome dependent mostly on class that has the highest likelihood to result in the response. To predict an outcome classification determined by the largest vote total, it essentially averages information findings of every classification that was provided through into voting classification. This objective seems to be to build a single prototype that train from these approaches but also estimates approach that relies off its accumulated majorities in voting with each output variable, instead than developing standalone algorithms as well as determining their accurateness of everyone.

4.2. Statistical metrics

In this study, we employ three statistical metrics that evaluate the outcomes from the aforementioned techniques and demonstrates optimal performance.

- Precision score:
  It is also called as “Positive predictive value (PPV)”. This is derived from the ratio of correct forecasts to all those provided by even the approach. The optimal precision for such good classification was 1. Only when the numerator as well as denominator were equivalent will precision become 1. FP must be 0 as well, when TP= TP + FP. Denominator value will increase with FP, passing numerator value, whereas precision value drops. In Eq. (1), where “TP” is considered as True positive, “FP” is derived as false positive.

$$PPV=TP/[TP+FP] \qquad (1)$$

- Recall score:
  Recall Eq. (2) is determined in such a classifier involving two categories via dividing the total number on true positives by the total of false negatives and true positives.

$$TPR=TP/[TP+FN] \qquad (2)$$

- F1-score:
  The weighted mean for recall and precision becomes a F1 score. Both false positive and false negative outcomes may occur with precision and recall, as is well known, thus both are taken into account. Eq. (3) defined as follows:

$$F1\ score=(2\cdot PPV\cdot TPR)/(PPV+TPR)$$

$$= (2\cdot TP)/(2\cdot TP+FP+FN) \quad (3)$$

5. Results & Analysis

We therefore assess our technique's outcomes as well as evaluate it to earlier prediction approaches [3], [5] and [9] that other researchers have suggested.

Here we examine job scheduling failure predicting initially. Relying upon these parameters as well as performance scores, categorized the status for activity inputs at the job level. Every status accomplished was taken into account into one category across all target categories, while the status failure is taken into account like another category.

Table 3. Accuracy score for models

Performance	
Models	**Accuracy Score**
Random Forest (RF)	**94.625**
Logistic Regression (LR)	92.5
Decision Tree (DT)	90.85
Naïve Bayes (NB)	92.5
KNN	94.47
SVM	92.5
Voting Classifier (VC)	94.55

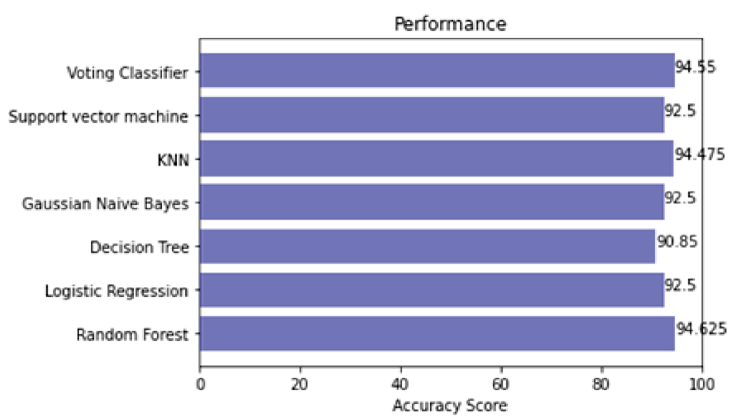

Fig. 4. Graphical illustration of accuracy scores

Overall accuracy score for every algorithm is classified their status on the basis of the dataset is demonstrated in Table 3. Every machine learning technique performance evaluation are apparent and efficient. Overall highest accuracy, especially, is 94.72% for the Random Forest approach. Each performance of accuracy score is graphically illustrated in above Fig. 4. The outcome is as follows RF > VC > KNN > NB > LR > SVM > DT. The Precision score for the Random Forest is the optimal which is of the 0.89. Precision, recall and F1 scores for the job failure status has been demonstrated into parts of success and failed segments. The tabular and graphical representation of precision score for overall machine learning approaches are displayed above in Table.4 and Fig. 5.

Table 4. Precision score for models

Models	Precision score Task/Job Status 0 (Success)	1 (Fail)	Average
RF	0.95	0.83	0.89
LR	0.93	0.0	0.46
DT	0.95	0.40	0.68
NB	0.93	0.0	0.46
KNN	0.95	0.78	0.87
SVM	0.93	0.0	0.46
VC	0.95	0.82	0.88

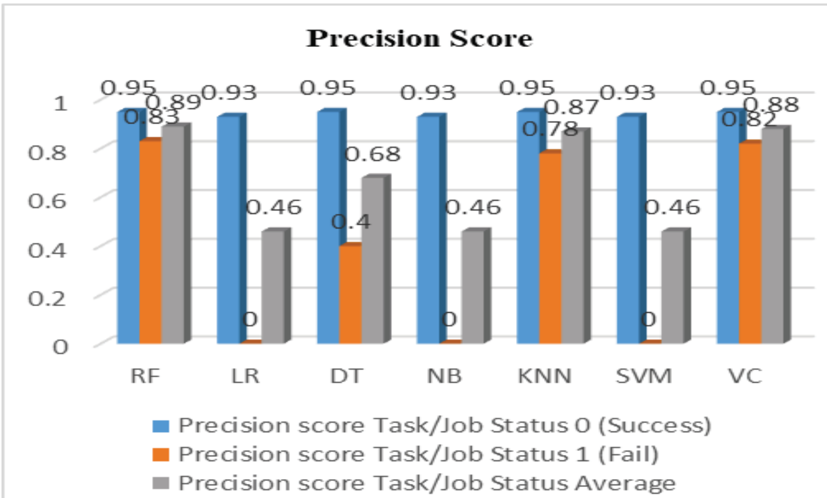

Fig. 5. Illustration for precision score

Random Forest approach of machine learning exhibited the higher F1 score with 0.74 and Decision tree achieved highest recall score with 0.69 about the job failures statuses. The below Table. 5 & Table. 6 shows the recall and F1 scores for every model involved in this study.

Table 5. Recall scores for ML models

Models	Recall score Task/Job Status 0 (Success)	1 (Fail)	Average
RF	0.99	0.37	0.68
LR	1.00	0.0	0.50
DT	0.95	0.44	0.69
NB	1.0	0.0	0.50
KNN	0.99	0.36	0.68
SVM	1.0	0.0	0.50
VC	0.99	0.36	0.68

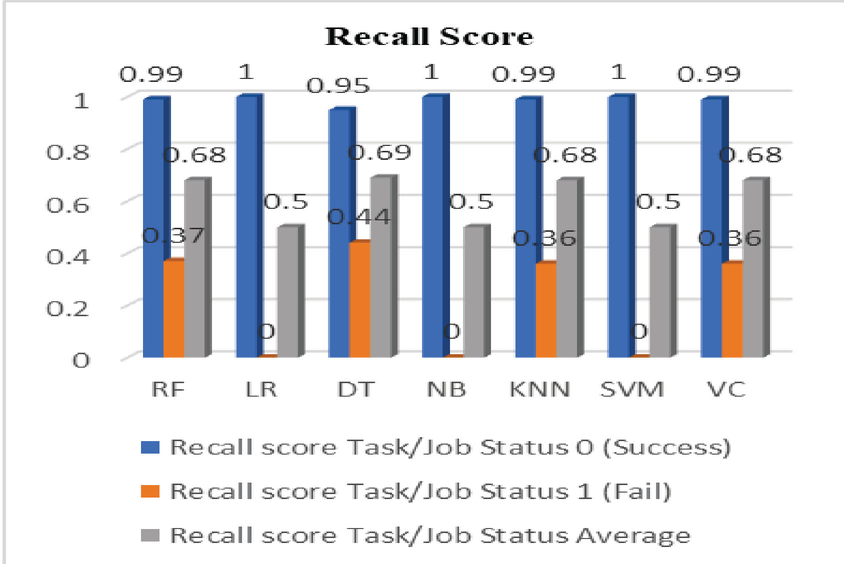

Fig. 6. Illustration of recall scores

By the Fig. 6 and Fig. 7. illustrate a Recall and F1 scores of each approach for job failure activity with various input sizes.

Table 6. F1-scores for models

F1-score			
	Task/Job Status		
Models	**0 (Success)**	**1 (Fail)**	**Average**
RF	0.97	0.51	0.74
LR	0.96	0.0	0.48
DT	0.95	0.42	0.68
NB	0.96	0.0	0.48
KNN	0.97	0.50	0.73
SVM	0.96	0.0	0.48
VC	0.97	0.50	0.73

The DT technique recorded the second-highest accuracy. The accuracy differential between the VC and KNN methods has been marginal gap, according to the conclusions. While testing on the dataset, the VC model had an accuracy of 94.57% and KNN achieved 94.47% accuracy. In metrics of precision score, second higher score acquired is RF and VC with 0.88. RF, VC and KNN has the second highest recall score with 0.68. In term of F1 score, VC and KNN achieved second highest score with 0.73. Finally, RF and VC approaches built for this job failure analysis, executed adequately in every statistical metrics involved in this study.

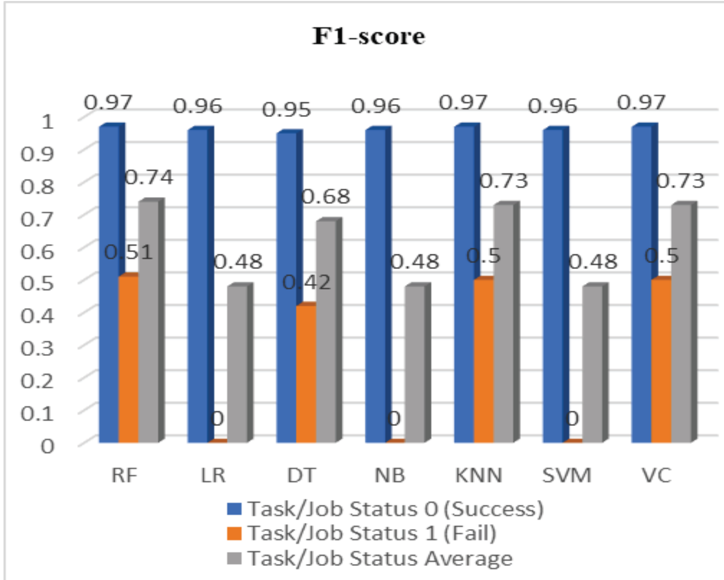

Fig. 7. Illustration of F1 scores

6. Conclusion

Adequate reliability and task fail predictions are vital to application quality of services in cloud data centers. In this study, we imply a rigorous interpretivist paradigm for prediction of task failure on Google cluster dataset. Due to this, we built machine learning models and examined the performance based on the few statistical metrics. According to the simulation results, the Random Forest classifier seems to be the optimal model for predicting the job failure activity. RF has accomplished an accuracy of 94.625%, precision score of 0.89 and F1 score of 0.74. In term of recall score with 0.69, DT achieved optimal solution. Ultimately, this research indicates that algorithms based on machine learning performed well in terms of evaluating overall status of job failure prediction.

Acknowledgement & Conflict of interest

"The article does not contain any studies with human participants or animals performed by any of the authors. Both authors declare, has no conflict of Interest".

References

1. E. W. J. W. S. D. L. O. S. a. E. E. M. Sedaghat, "DieHard: Reliable Scheduling to Survive Correlated Failures in Cloud Data Centers," Cloud and Grid Computing (CCGrid),, pp. 52-59, 2016.
2. R. P. M.-R. R. a. S. B. Subrata Mitra, "Partial-parallel-repair (PPR): a distributed technique for repairing erasure coded storage," Association for Computing Machinery, p. 1–16, 2016.
3. H. W. a. H. S. J. Gao, "Machine Learning Based Workload Prediction in Cloud Computing," International Conference on Computer Communications and Networks (ICCCN), pp. 1-9, 2020.
4. J. J. D. a. T. F. T. -P. Pham, "Predicting Workflow Task Execution Time in the Cloud Using A Two-Stage Machine Learning Approach," IEEE Transactions on Cloud Computing, vol. 8, pp. 256-268, 2020.
5. H. W. a. H. S. J. Gao, "Task Failure Prediction in Cloud Data Centers Using Deep Learning," IEEE International Conference on Big Data (Big Data),, pp. 1111-1116, 2019.
6. J. H. Y. S. C. L. B. C. a. J. C. C. Liu, "Predicting of Job Failure in Compute Cloud Based on Online Extreme Learning Machine: A Comparative Study," IEEE Access, vol. 5, pp. 9359-9368, 2017
7. P. U. A. Padmakumari, "Task Failure Prediction using Combine Bagging Ensemble (CBE) Classification in Cloud Workflow," Wireless Personal Communications, p. 23–40, 2019.
8. Q. H. M. Mohammad S. Jassas, "Analysis of Job Failure and Prediction Model for Cloud Computing Using Machine Learning," Sensors (Basel), 2022.

9. T. I. A. &. S. J. Tengku Asmawi, "Cloud failure prediction based on traditional machine learning and deep learning," Journal of Cloud Computing, 2022.
10. L. D. Y. L. G. L. &. W. M. Chunhong Liu, "Failure prediction of tasks in the cloud at an earlier stage: a solution based on domain information mining," Computing, 2020
11. R. S. a. S. G. J. Shetty, "Task Resource Usage Analysis and Failure Prediction in Cloud," Data Science & Engineering (Confluence), pp. 342-348, 2019.
12. T. I. a. D. Manivannan, "Predicting Application Failure in Cloud: A Machine Learning Approach," IEEE International Conference on Cognitive Computing (ICCC), pp. 24-31, 2017.
13. T. D. a. A. A. Y. Alahmad, "Proactive Failure-Aware Task Scheduling Framework for Cloud Computing," IEEE Access, vol. 9, pp. 106152-106168, 2021.
14. Z. M. (. J. H. L. A. E. H. C. H. R. H. Z. Z. M. W. a. P. C. Yangguang Li, "Predicting Node Failures in an Ultra-Large-Scale Cloud Computing Platform: An AIOps Solution," ACM Transactions on Software Engineering and Methodology, vol. 29, no. 2, pp. 1-24, 2020.
15. S. J. M. K. S. H. C. S. K. K. I. B. Rakesh Kumar, "The Mystery of the Failing Jobs: Insights from Operational Data from Two University-Wide Computing Systems," 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 158-171, 2020.
16. S. P. a. D. P. D. Basak, "Support vector regression," Neural Information Processing—Letters and Reviews, 2007
17. M. W. Tolles J, "Logistic Regression: Relating Patient Characteristics to Outcomes.," JAMA Guide to Statistics and Methods, 2016.
18. S. H. S. D. Zhu X, "A novel matrix-similarity based loss function for joint regression and classification in AD diagnosis," Neuroimage, 2014.
19. S. Xu, "Bayesian Naïve Bayes classifiers to text classification," Journal of Information Science, p. 48–59, 2018.
20. D. K. M. M. Saloni Kumari, "An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier," International Journal of Cognitive Computing in Engineering, vol. 2, pp. 40-46, 2021.