ISSN: 2321-7782 (Online)
e-ISJN: A4372-3114
Impact Factor: 7.327

Volume 6, Issue 6, June 2018
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Exploratory Data Analysis on Hard Drive Failure Statistics and Prediction

Shivam Bhardwaj[1]
Department of Computer Engineering
Bharati Vidyapeeth Deemed University
College of Engineering
Pune – India

Akshay Saxena[2]
Department of Computer Engineering
Bharati Vidyapeeth Deemed University
College of Engineering
Pune – India

Achal Nayyar[3]
Department of Computer Engineering
Bharati Vidyapeeth Deemed University
College of Engineering
Pune – India

Abstract: Data centers generally use hard drives as data storage device. Large companies heavily rely on data and use many hard drives, which become challenging to monitor manually. When there is an issue with the hard-disk, it should function for at least next 24 hours for the data back-up to be done. But in ideal cases, the hard-disk fails even before 24 hours resulting in loss of data. Hard drive failures cause data loss which can cause a serious problem for the users. As backup, multiple copies of data can be stored in the system, but it might increase the cost at the same time. In industries, hard drives are monitored by setting threshold for several critical metrics.

SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes of hard disks can be useful in detecting the failure rate of the hard-disks. It is useful to predict the hard drive failure by developing a model, so that it can be used to get useful insights to improve the system reliability and help cut cost.

Keywords: Hard Drive Failure Prediction; Data Center Predictive Maintenance; Artificial Neural Network; Ensemble Modelling; Feature Importance.

I. INTRODUCTION

Disk failures are not rare in Datacenter and cloud computing environments. Fortunately, we have S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology; often written as SMART) logs collected from computer hard disk drives (HDDs), Solid-State Drives (SSDs) and eMMC drives that detects and reports on various indicators of drive reliability, with the intent of enabling the anticipation of hardware failures. Hence, HDD vendors are highly motivated to reduce the rate of failures as a cost saving measure. SMART attributes represent HDD health statistics such as the number of scan errors, real location counts and probational counts of a HDD. If a certain attribute considered critical to HDD health goes above its threshold value, the HDD is marked as likely to fail. This report focuses on applying machine learning to improve prediction accuracy over heuristics in hard disk drives.

II. DATA DESCRIPTION

A. Data Source

The dataset under consideration is hard drive dataset, published by Backblaze. Backblaze records SMART stats of 67,814 hard drives, which are running every day in their Sacramento data center. SMART stands for Self-Monitoring, Analysis and Reporting Technology, is a monitoring system included in hard drives to report attributes about a given drive. The data for the following experiment in a cleaner format can be found on Kaggle:

https://www.kaggle.com/backblaze/hard-drive-test-data

The original source of data is:

https://www.backblaze.com/b2/hard-drive-test-data.html

B. Data Summary

Each day a snapshot of each operational hard drive is taken in the Backblaze data center. The snapshot will have the basic drive information along with the SMART statistics reported by that drive. The daily snapshot of one drive is one record or row of data. The snapshots of the drives are compiled into a single file which further consists of separate row to which denotes the status of the hard drive. The detailed description of dataset is as follows.

Date – The date of the file in yyyy-mm-dd format.

Serial Number – The manufacturer-assigned serial number of the drive.

Model – The manufacturer-assigned model number of the drive.

Capacity – The drive capacity in bytes.

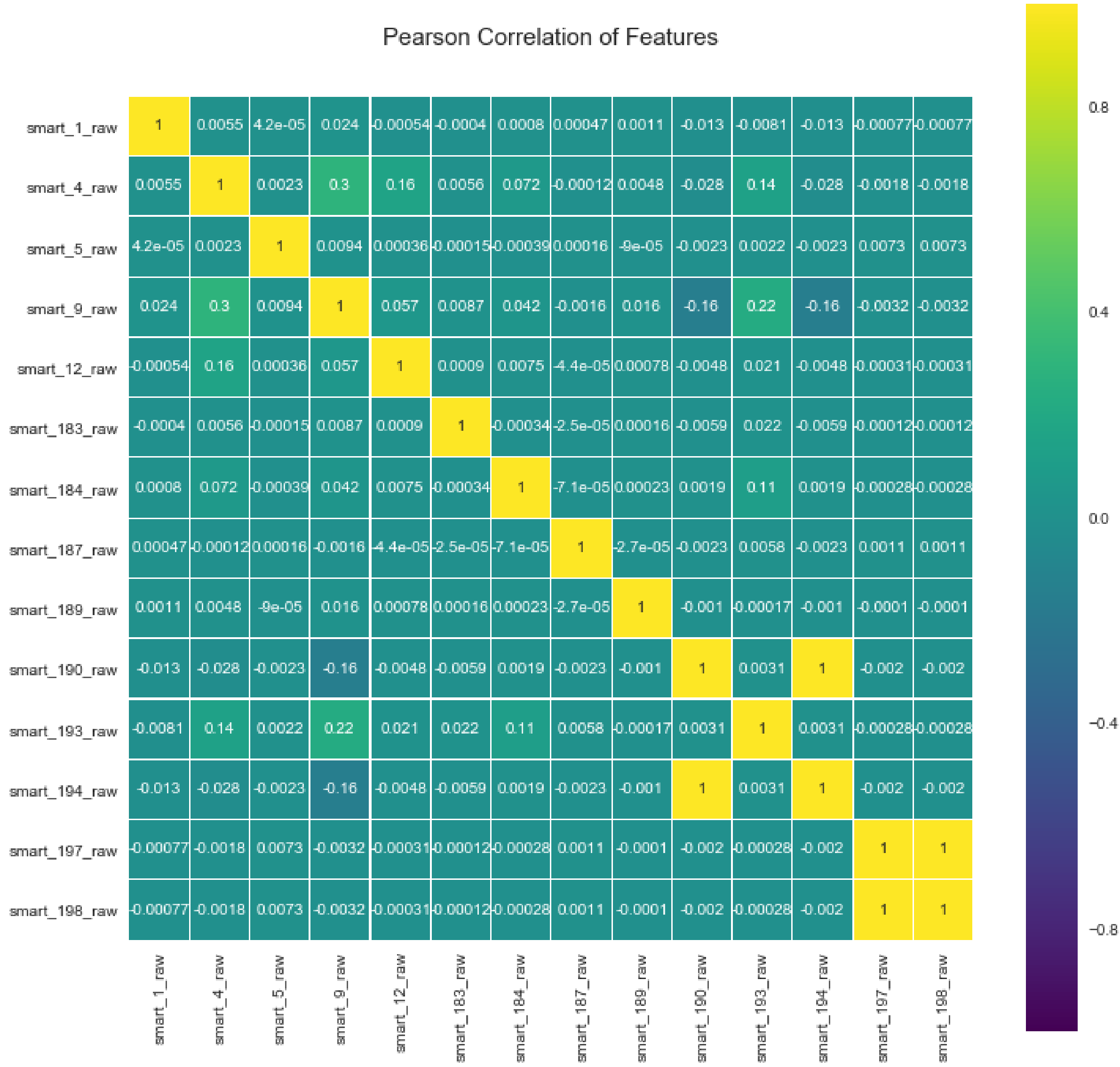
Failure – Contains a "**0**" if the drive is OK. Contains a "**1**" if this is the last day the drive was operational before failing.

SMART Stats – 95 columns of data, that are the Raw and Normalized values for 45 different SMART stats as reported by the given drive. Each value is the number reported by the drive.

III. Exploratory Data Analysis

The full version of the dataset comprises of day-wise observations covering ~ 67,814 hard drives over the span of Jan 2015 – Dec 2017. We consider taking Seagate model number ST4000DM000 as a subset for the analysis. This subset shall be henceforth referred to simply as the dataset in this document. The following is the correlation matrix generated using the SMART raw values: [Figure 3: Pearson - Correlation Matrix] Inferences drawn from the correlation matrix are as follows:

- SMART 4 and 192 exhibit high correlation as they relate to the number of cycles on start after shutdown. 192 captures power off cycles and is complemented by 4 which increments the value on startup.
- SMART 190 and 194 deal with temperature, hence highly correlated.
- SMART 197 and 198 exhibit high correlation because 197 defines unstable sectors due to read errors and 198 gives count of uncorrectable errors while read/write to a sector. We take 198 and ignore 197.
- SMART 9,12 are correlated to an extent as they cover related features - number of hours the drive is up, count of full power on/off cycles, and the Logical Block Addresses read during the time it was up.



[Figure 2: Pearson Correlation Matrix]

IV. Modeling

A. Selecting the Model

The following modeling techniques were used to predict the hard disk failure using the extracted features:

• Random Forest

• Fully connected Artificial Neural Network

• XGboost

Few features we considered before using the listed models:

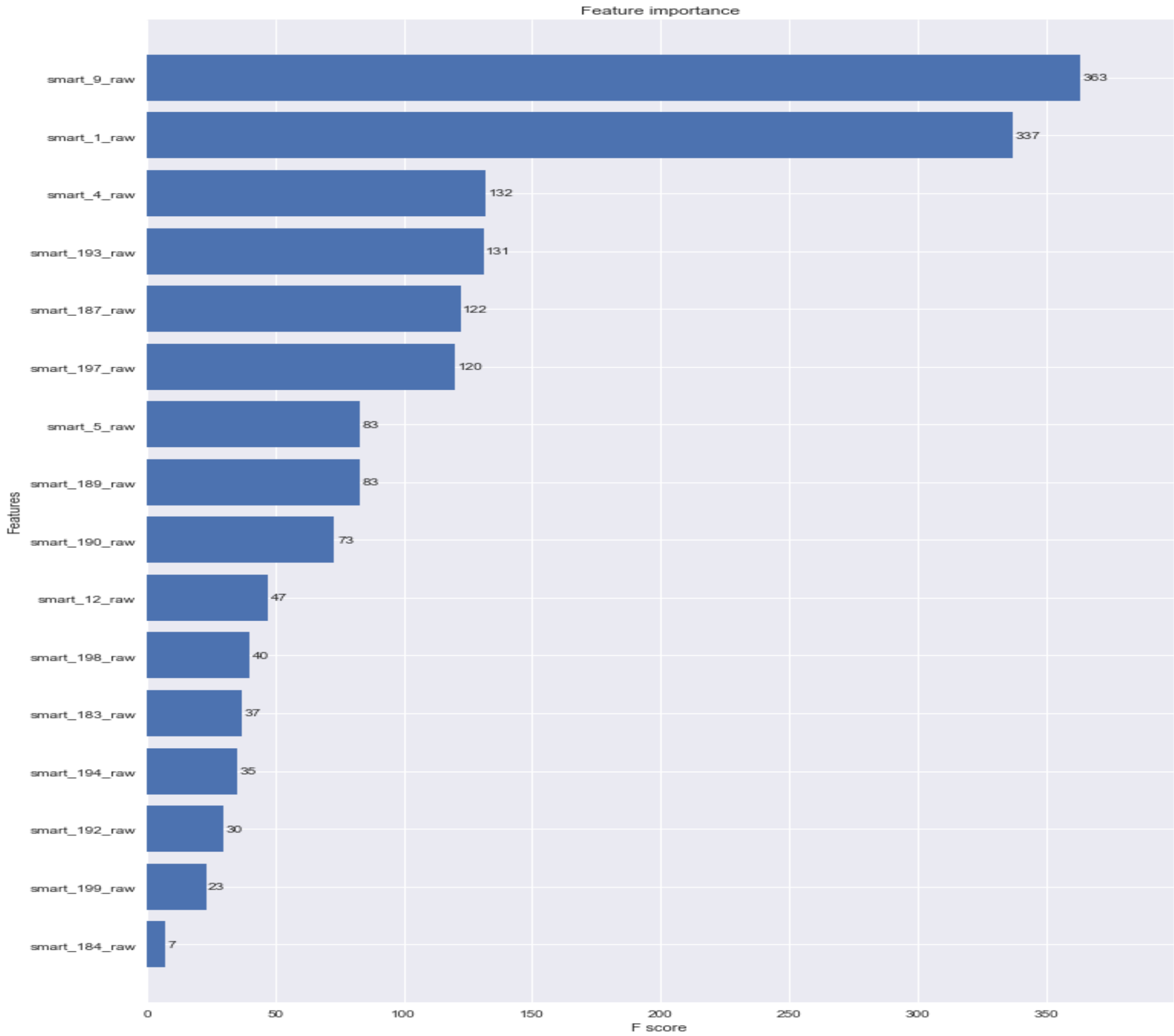
In our given problem it is easy to interpret and straightforward to visualize, this will help us to explain it to business. - One important caveat in using the decision tree model is that it tries to make an optimal prediction at every node level. This makes it prone to overfitting, especially when it is deep. This is due to the amount of specificity at each node level. To avoid this pitfall, we build a random forest to compare with it. - After a thorough cleaning and applying principal component analysis, the data set was set up for Artificial Neural Network and Extreme Boosting Algorithm(XgBoost) as they performed well even on a very imbalanced anomaly detection.

B. Building the Model

The dataset was split into train and test sets based on the test design. For each trial, the records were sampled using stratified random sampling and trained on each of the selected model. 5-fold cross validation was the evaluation criteria before the final test prediction. PCA was used as the dimensionality reduction technique and the Principal Components were generated for each validation set using the coefficients in each of the cross-validation fold. The same approach was also considered for the final test sets. The experiment was setup on Jupyter Lab with GPU support. Also used Google Colaboratory using Keras Library on Tensorflow backend.

The [Figure 2: Feature Importance] depicted by Running XGBoost estimator, the following features were the only ones listed (in order of importance)

Smart 9, Smart 1, Smart 197, Smart 187, Smart 4, Smart 193, Smart 198, Smart 190, Smart 189, Smart 5, Smart 12, Smart 199, Smart 194, Smart 192, Smart 183, Smart 184.



[Figure 1: Feature Importance]

V. Model Performance

Depending on the type of output the model creates we assess them differently. As assessing a model is an integral part of model building these parameters were taken into consideration:

Confusion Matrix: It is an N*N matrix, where N denotes no of classes predicted. From the confusion matrix we derive these sub-metrics:

Accuracy - % of correct predictions

Kappa - a metric that compares an Observed Accuracy with an Expected Accuracy

Precision - % of positive cases which were correct

Sensitivity/Recall - a portion of actual positive cases which were predicted correctly Specificity - a portion of actual negative cases which were predicted correctly.

ROC and Area under the Curve - A plot of TPR vs FPR to showcase the strength of the model.

All the model evaluation metrics are shown in [Figure 3: Model Performance Evaluation], One of the most important metric for the experiment being

Fmeasure - % of correct predictions from for Type II error.

		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive,** Power	**False positive,** Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative,** Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$	F_1 score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$		

[Figure 3: Model Performance Evaluation Matrix]

The models trained on the new feature set gave better results than the models trained on just last day's data. Amongst the three models trained the XGboost and Neural Network models gave the most optimal accuracy and sensitivity. But considering the time needed to train the model, the bagging based algorithm was faster to train.

VI. Conclusion

In this experiment, we have analysed the Backblaze hard drive (Seagate - ST4000DM000) failure and used several prediction models for classification. We evaluated the prediction performance among Random Forest, Gradient Boosting (XGBoost) and Artificial Neural Network (ANN) models. The data set was highly imbalance in nature also the SMART statistics alone cannot provide the best model as the other constraints for hard disk failure are numerous and not all constraints can be monitored. The false negative rate was high due to those failed hard drives not having any relevant SMART data indicators for failure.

Total number of drives	11569
Number of failed drives	42
Number of predicted failures	25
Number of false positives	09
Number of false negatives	26
Number of true negatives	00
Number of true positives	16
Fmeasure	0.733944954128

[Figure 4: Model Performance Evaluation Results]

In this report we have limited our scope to only Seagate model number ST4000DM000, our analysis and prediction can be further extended to other hard drive models from Backblaze. Furthermore, we are now predicting the hard drive device failure on the day of its failure. If we could predict the device failure in advance, then suitable backup action can be taken to avoid the data loss. Training time of Fully connected Neural Network can be reduced by building efficient RNN and LSTM networks that can also self-analyse and predict the hard drive failure in advance.

References

1. E. Pinheiro, W.-D. Weber, L. A. Barroso, "Failure trends in a large disk drive population", Proceedings of the 5th USENIX Conference on File and Storage Technologies FAST '07, USENIX Association, pp. 17-29, 2007.
2. M. M. Botezatu, I. Giurgiu, J. Bogojeska, D. Wiesmann, "Predicting disk replacement towards reliable data centers", Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD '16, ACM, pp. 39-48, 2016.
3. Ceglar, J.F Roddick. "Association mining". ACM Computing Surveys, 38:2, pp. 1-42, 2006.
4. Chmielewski and Grzymala-Busse Global discretization of continuous attributes as pre-processing for machine learning. In Third International Workshop on Rough Sets and Soft Computing 1994, pp. 294–301.
5. Dougherty et al., Supervised and unsupervised discretization of continuous features. In Proc. Twelfth International Conference on Machine Learning. Los Altos, CA: Morgan Kaufmann, 1995 pp. 194– 202.
6. Han, J., Hu, X., Lin, T. Y. (2004). Feature Subset Selection Based on Relative Dependency between Attributes. Rough Sets and Current Trends in Computing: 4th International Conference, RSCTC 2004, Uppsala, Sweden, June 1-5, pp. 176–185.
7. Jensen R. and Shen Q (2005). Semantics-Preserving Dimensionality Reduction: Rough and Fuzzy-Rough-Based Approaches, IEEE Transactions on Knowledge And Data Engineering, Vol. 17, No. 1.
8. Kotsiantis S. and Kanellopoulos. D. "Association Rules Mining" A Recent Overview. GESTS Int. Transactions on Computer Science and Engineering, Vol. 32 (1), pp. 71-82, 2006.
9. Liu H. et al., Discretization: An Enabling Technique, Data Mining and Knowledge Discovery, 6, 393–423, 2002. Kluwer Academic Publishers, the Netherlands.
10. Mehmed Kantardzic, Data Mining: Concepts, Models, Methods, and Algorithms 2003 John Wiley & Sons publishers.
11. Nguyen H. S. and Skowron, Quantization of real value attributes: rough set and Boolean reasoning approach. Proceedings of the Second Joint Annual Conference on Information Sciences, pp. 34-37, Wrightswille Beach, NC, September 1995
12. Archana, R. C., Naveenkumar, J., & Patil, S. H. (2011). Iris Image Pre-Processing And Minutiae Points Extraction. International Journal of Computer Science and Information Security, 9(6), 171.
13. Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2016). A Survey on the Anomalies in System Design: A Novel Approach. International Journal of Control Theory and Applications, 9(44), 443–455.
14. Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2017a). A Stochastic Software Development Process Improvement Model To Identify And Resolve The Anomalies In System Design. Institute of Integrative Omics and Applied Biotechnology Journal, 8(2), 154–161.
15. Bhore, P. R., Joshi, S. D., & Jayakumar, N. (2017b). Handling Anomalies in the System Design: A Unique Methodology and Solution. International Journal of Computer Science Trends and Technology, 5(2), 409–413.
16. Desai, P. R., & Jayakumar, N. K. (2017). A Survey on Mobile Agents. International Journal for Research in Applied Science & Engineering Technology (IJRASET), 5(XI), 2915–2918.
17. Gawade, M. S. S., & Kumar, N. (2016). Three Effective Frameworks for semi-supervised feature selection. International Journal of Research in Management & Technology, 6(2), 107–110.
18. GAWADE, S., & JAYKUMAR, N. (2017). ILLUSTRATION OF SEMI-SUPERVISED FEATURE SELECTION USING EFFECTIVE FRAMEWORKS. Journal of Theoretical & Applied Information Technology, 95(20).
19. Jaiswal, U., Pandey, R., Rana, R., Thakore, D. M., & JayaKumar, N. (2017). Direct Assessment Automator for Outcome Based System. International Journal of Computer Science Trends and Technology (IJCS T), 5(2), 337–340.
20. Jayakumar, D. T., & Naveenkumar, R. (2012). SDjoshi, International Journal of Advanced Research in Computer Science and Software Engineering, Int. J, 2(9), 62–70.
21. Jayakumar, M. N., Zaeimfar, M. F., Joshi, M. M., & Joshi, S. D. (2014). INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET). Journal Impact Factor, 5(1), 46–51.
22. Jayakumar, N. (2014). Reducts and Discretization Concepts, tools for Predicting Student's Performance. International Journal of Engineering Science and Innovative Technology (IJESIT), 3(2), 7–15.
23. Jayakumar, N. (2015). Active storage framework leveraging processing capabilities of embedded storage array.
24. Jayakumar, N., Bhardwaj, T., Pant, K., Joshi, S. D., & Patil, S. H. (n.d.). A Holistic Approach for Performance Analysis of Embedded Storage Array.
25. Jayakumar, N., Iyer, M. S., Joshi, S. D., & Patil, S. H. (2016). A Mathematical Model in Support of Efficient offloading for Active Storage Architectures. In International Conference on Electronics, Electrical Engineering, Computer Science (EEECS) : Innovation and Convergence (Vol. 2, p. 103).
26. Jayakumar, N., & Kulkarni, A. M. (2017). A Simple Measuring Model for Evaluating the Performance of Small Block Size Accesses in Lustre File System. Engineering, Technology & Applied Science Research, 7(6), 2313–2318.
27. Jayakumar, N., Singh, S., Patil, S. H., & Joshi, S. D. (n.d.). Evaluation Parameters of Infrastructure Resources Required for Integrating Parallel Computing Algorithm and Distributed File System.
28. KAKAMANSHADI, M. G., NAVEENKUMAR, M. J., & PATIL, S. H. (2011). A METHOD TO FIND SHORTEST RELIABLE PATH BY HARDWARE TESTING AND SOFTWARE IMPLEMENTATION. International Journal of Engineering Science.
29. Kulkarnia, A., & Jayakumar, N. (2016). A Survey on IN-SITU Metadata Processing in Big Data Environment. International Journal of Control Theory and Applications, 9(44), 325–330.
30. Kumar, N., Angral, S., & Sharma, R. (2014). Integrating Intrusion Detection System with Network Monitoring. International Journal of Scientific and Research Publications, 4, 1–4.
31. Kumar, N., Kumar, J., Salunkhe, R. B., & Kadam, A. D. (2016). A Scalable Record Retrieval Methodology Using Relational Keyword Search System. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies(p. 32).
32. kumar Singha, A., Patilb, S. H., & Jayakumar, N. (2017). A Treatment for I/O Latency i n I/O Stack. Http://Www.Ijcstjournal.Org/Volume-5/Issue-2/IJCST-V5I2P83.Pdf.
33. Namdeo, J., & Jayakumar, N. (2014). Predicting Students Performance Using Data Mining Technique with Rough Set Theory Concepts. International Journal of Advance Research in Computer Science and Management Studies, 2(2).
34. Naveenkumar, J. (2012). Keyword Extraction through Applying Rules of Association and Threshold Values. International Journal of Advanced Research in Computer and Communication Engineering, 1(5), 295–297. Retrieved from http://www.ijarcce.com/upload/july/3-Keyword Extraction.pdf
35. Naveenkumar, J., & Joshi, S. D. (2015). Evaluation of Active Storage System Realized Through Hadoop. International Journal of Computer Science and Mobile Computing, 4(12), 67–73.
36. Naveenkumar, J., Makwana, R., Joshi, S. D., & Thakore, D. M. (2015a). OFFLOADING COMPRESSION AND DECOMPRESSION LOGIC CLOSER TO VIDEO FILES USING REMOTE PROCEDURE CALL. Journal Impact Factor, 6(3), 37–45.
37. Naveenkumar, J., Makwana, R., Joshi, S. D., & Thakore, D. M. (2015b). Performance Impact Analysis of Application Implemented on Active Storage Framework. International Journal, 5(2).
38. Naveenkumar, J., & Raval, K. S. (2011). Clouds Explained Using Use-Case Scenarios. In INDIACom-2011 Computing For Nation Development (pp. 1–5).
39. Naveenkumar J, P. D. S. D. J. (2015). Evaluation of Active Storage System Realized through MobilityRPC. International Journal of Innovative Research in Computer and Communication Engineering, 3(11), 11329–11335.
40. NAVEENKUMAR, M. J., Bhor, M. P., & JOSHI, D. R. S. D. (2011). A Self Process Improvement For Achieving High Software Quality. International Journal of Engineering Science,
41. RAVAL, K. S., SURYAWANSHI, R. S., NAVEENKUMAR, J., & THAKORE, D. M. (2011). The Anatomy of a Small-Scale Document Search Engine Tool: Incorporating a new Ranking Algorithm.
42. Rishikesh Salunkhe, N. J. (2016). Query Bound Application Offloading: Approach Towards Increase Performance of Big Data Computing. Journal of Emerging Technologies and Innovative Research, 3(6), 188–191.
43. Salunkhe, R., Kadam, A. D., Jayakumar, N., & Joshi, S. (n.d.). Luster A Scalable Architecture File System: A Research Implementation on Active Storage Array Framework with Luster file System.
44. Salunkhe, R., Kadam, A. D., Jayakumar, N., & Thakore, D. (n.d.). In Search of a Scalable File System State-of-the-art File Systems Review and Map view of new Scalable File system. In nternational Conference on El ectrical, Electronics, and Optimization Techni ques (ICEEOT) - 2016 (pp. 1–8).
45. Sawant, Y., Jayakumar, N., & Pawar, S. S. (2016). Scalable Telemonitoring Model in Cloud for Health Care Analysis. In International Conference on Advanced Material Technologies (ICAMT) (Vol. 2016).
46. Singh, A. K., Pati, S. H., & Jayakumar, N. (2017). A Treatment for I/O Latency in I/O Stack. International Journal of Computer Science Trends and Technology (IJCS T), 5(2), 424–427.
47. Zaeimfar, S. D. J. N. J. F. (2014). Workload Characteristics Impacts on file System Benchmarking. Int. J. Adv, 39–44.