An Approach to Failure Prediction in a Cloud Based Environment

Conference Paper · August 2017

DOI: 10.1109/FiCloud.2017.56

6 authors, including:

Hussaini Adamu
University of Bradford

Bashir Mohammed
Lawrence Berkeley National Laboratory

Ali Maina Bukar
University of Bradford

Andrea J. Cullen
University of Bradford

An approach to failure prediction in a cloud based environment

Hussaini Adamu†, Bashir Mohammed, Ali Bukar Maina, Andrea Cullen, Hassan Ugail, Irfan Awan,
Faculty of Engineering and Informatics, University of Bradford
Bradford, BD7 1DP, UK
{A.Hussaini239, B.Mohammed1, A.M.Bukar, A.J.Cullen, H.Ugail, I.U.Awan}@bradford.ac.uk

***Abstract*—Failure in cloud system is defined as an even that occurs when the delivered service deviates from the correct intended service. As the cloud computing systems continue to grow in scale and complexity, there is an urgent need for cloud service providers (CSP) to guarantee a reliable on-demand resource to their customers in the presence of faults thereby fulfilling their service level agreement (SLA). Component failures in cloud systems are very familiar phenomena. However, large cloud service providers' data centers should be designed to provide a certain level of availability to the business system. Infrastructure-as-a-service (Iaas) cloud delivery model presents computational resources (CPU and memory), storage resources and networking capacity that ensures high availability in the presence of such failures. The data in-production-faults recorded within a 2 years period has been studied and analyzed from the National Energy Research Scientific computing center (NERSC). Using the real-time data collected from the Computer Failure Data Repository (CFDR), this paper presents the performance of two machine learning (ML) algorithms, Linear Regression (LR) Model and Support Vector Machine (SVM) with a Linear Gaussian kernel for predicting hardware failures in a real-time cloud environment to improve system availability. The performance of the two algorithms have been rigorously evaluated using K-folds cross-validation technique. Furthermore, steps and procedure for future studies has been presented. This research will aid computer hardware companies and cloud service providers (CSP) in designing a reliable fault-tolerant system by providing a better device selection, thereby improving system availability and minimizing unscheduled system downtime.**

Keywords— Failure; Cloud Computing; Machine Learning; Availability.

I. INTRODUCTION

Increasing amount of cloud resources provide the infrastructure of ICT utilities at a global proportion. Cloud users request for cloud resources from Cloud service providers (CSP) to provide diverse ICT utilities such as business-critical processes, high performance computing, social networking and scientific computing. Due to the sheer scale of cloud datacenters, resources failure are inevitable and bound to happen, therefore it is of critical importance to ensure the reliability and availability in such systems. There is also an urgent need for CSP to offer a scalable, efficient and reliable on-demand resource to their customers in the presence of faults thereby fulfilling their service level agreement (SLA). Component failures within the cloud infrastructure are common, but large cloud datacenters should be designed to guarantee a certain level of availability to the Business system. Infrastructure-as-a-Service (IaaS) cloud presents computational resources (e.g., CPU and memory), storage resources, and networking capacity that ensures high availability in the face of such failures[1]. Cloud systems can have tremendous failure rates as they feature many servers that are geographically dispersed with a high workload. The availability of such systems can be quickly endangered if the failure is not sufficiently handled[2]. To guarantee availability of services to cloud users, cloud infrastructures should be designed such that they should have minimal or insignificant system downtime. Replication of data and checkpointing technique are some of the common existing strategies used to ensure availability of cloud services[3].

Failure prediction is necessary for predictive maintenance due to its ability to prevent failure incidents and maintenance costs[4]. Predictive maintenance is about anticipating failures and taking proactive actions[5]. Recent advances in machine learning and cloud storage have created a great opportunity to utilize the huge amount of data generated from cloud infrastructures which provides room to predict when a component is likely to malfunction or fail. Currently, mathematical and statistical modeling are the prominent approaches used for failure predictions, these are based on equipment degradation physical models and machine learning techniques, respectively[6]. According to [7], Cloud computing is usually associated with failures. The risk of failure can be viewed as the possibility of suffering loss, or exposure in the cloud-computing life cycle. Generally, cloud computing risk management consists of processes, approaches, and techniques that are employed to reduce cloud computing risks failure. Although, much research and advancement have been carried out in this area cloud, some companies have suffered a huge amount of downtime as a result of cloud failure which has led to a significant revenue loss [7]. Some instances of cloud failures are the Database Cluster failure caused at Saleforce.com. Also in 2011 Microsoft Cloud service outage lasted for 2.5 hours[8], with Google Docs service outage lasting for an hour. These were because of memory leaks due to

978-1-5386-2074-8/17 $31.00 © 2017 IEEE
DOI 10.1109/FiCloud.2017.56

a software update [9], [10], costing both business millions of dollars. Similar reports were witnessed by Gmail services down for about 50 minutes, Amazon Web services for 6 hours, while Facebook's photos and "likes" services were down costing customer satisfaction. Multiple business hosting their websites, such as with GoDaddy, suffered 4 hours' downtime affecting 5 million websites[10]. So having a pre-knowledge of the failures emerging within the cloud infrastructures will assist in minimizing the effect of cloud failures thereby preventing business and financial losses, even though according to some researchers there are possibilities that in the future, SLA-Based Google App engine would expect to manage all causes of failures[11]. The paper is organized as follows: Section 2 presents some related work while Section 3 briefly discusses the concept of cloud computing and its deployment models. Section 4 presents an overview of the NERSC data while Section 5 describes the methodology of our approach. Experiment and discussion of result is presented in Section 6 and finally Section 7 concludes the paper and suggests future work.

II. RELATED WORK

A large number of research effort have been devoted to improve the efficiency of several approaches and procedures in failure prediction[12],[13]–[16],[17],[6], [18] but very few have addressed the issue of failure prediction in a cloud based environment[4],[19],[17],[20]. We limit our review to recent research work conducted in this area. For instance the authors in [6] used Bayesian network to predict failure probabilities. While the research seamed interesting, they did not disclose the dataset used in conducting the analysis thus making it hard to replicate or compare other Machine Learning (ML) Algorithms to their proposed strategy. Authors in [19] used an ensemble classifier to achieve hard drive failure prediction on a cloud infrastructure. The data they conducted their work on was acquired through two sources, namely Windows performance counts and Self-Monitoring Analysis and Reporting Technology (S.M.A.R.T or SMART)[21]. This research closely resembles the intended work, but they only considered hard disk failure in the cloud architecture while real time business critical systems relies on other components and not only hard drive, but rather a host of Hardware (such as: CPU, Disk, DIMM, Cable .etc.).

Recently, authors in [18] used data acquired from cycles to predict Integrated Circuit (IC) failures. Same in the case of [19] they also considered only one Hardware failure occurrence. They analyzed fourteen (14) hardware samples which is quite impressive. However, the main limitation is that the data they used has not been made publicly available. Our approach is to use a publicly available hardware dataset to gain a machine leaning (ML) classifier to predict hardware failures, contrary to most of the state- of- the art research work being conducted in this area. Our choice of selecting a public dataset in performing our analysis is simply to enable other researchers in the field to compare their outcome with our obtained results. Furthermore, in this work we are not limiting our experiments to a single hardware, rather we attempt to predict several hardware failures within a cloud infrastructure. For more comprehensive review on other literatures or works by other scholars, the reader is referred to [22],[23] ,[24],[25],[26] ,[27],[28],[29],[5], [6], [30]

III. THE CONCEPT OF CLOUD COMPUTING

A. The Cloud Concept and it's Deployment Models

Cloud computing can be categorized into three distinct elements. Each element has a purpose which performs specific tasks as follows; data centers, distributed servers, and clients[27].The four main cloud deployment models are public, private, community and hybrid[31].

- Public clouds are provided to the public or a large industry group. This is managed by a third party selling cloud services. As cloud technology develops, public cloud services are becoming more attractive to Business companies as well. Many actors such as critical infrastructure providers, including financial institutions has shown special interest in cloud[32].

- Private cloud is owned and managed by a single organization that concentrates on controlling the mechanism of visualizing resources and automating services that are used and customized by various lines of business and constituent groups. A private cloud provides services to an organization through an intranet. Private clouds can be linked to each other to form a partner cloud. Private clouds are operated solely for an organization[7].

- Community cloud it is dedicated or allows services to be available to a professional community or group of organization that comprises of Subcontractors, Branches, allies and so forth to operate collaboratively on a project. It may also be a government cloud dedicated to state establishments [33].

- Hybrid cloud model is a mixture or combination of private and public cloud infrastructures working together. As a result, hybrid cloud inherits the properties of both private cloud and public cloud. It allows organizations to manage their critical data and applications in private while outsourcing other non-critical activities to the public cloud[34][35].

B. Cloud Service Models

There are three levels of essential services offered by cloud computing: Infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

- Infrastructure as a service (IaaS), is the most basic and important cloud service model under which virtual machines, load balancers, fault tolerance, firewalls and networking services are provided[36]. The client or cloud user, is provided with capability to provision processing, storage, networks and other fundamental computing resources, to deploy and run arbitrary software such as operating system and applications. Common examples of these services include Rackspace, GoGrid, EC2, Google Apps, Concur, Cisco Webex, Citrix GoTo Meetings, Adobe Marketing Cloud, Facebook, Flickr) and Amazon cloud[8] [37] .

- Under the PaaS model, a computing platform including APIs, operating system and development environments are provided as well as programming language execution environment and web servers. The client maintains the applications, while the cloud provider maintains the service run times, databases, server software, integrated server oriented architectures and storage networks. Various types of PaaS vendors offerings can include complete application hosting, development, testing and extensive integrated services that include scalability and maintenance[38]. Some key players include Microsoft Windows Azure and Google Apps engine GoDaddy, Windows Azure, Apprenda, Google App Engine, Amazon Web Services, and WordPress. The main benefit of these services include focus on high value software rather than infrastructure, leverage economies of scale and provide scalable go-to-market capability [39].

- SaaS provides clients the capability to use provider application executing on a cloud infrastructure. An entire application is available remotely and accessible from multiple client devices through thin client interfaces such as web browsers. Cloud user do not manage or control the underlying cloud infrastructure [2] but providers install and operate the application software. Example providers for this service include Salesforce, Facebook and Google Apps, Amazon EC2, Rackspace, Microsoft Azure, Google Compute Engine and Amazon Web Services [39]–[41].

IV. OVERVIEW OF THE NERSC DATA

This NERSC data [42] was collected with the purpose of providing failure specifics for I/O related systems and components in as much detail as possible so that analysis might produce some useful findings. Data were collected for storage, networking, computational machines, and file systems in production use at NERSC from the 2001-2006 timeframe. The data was extracted form a database used for tracking system troubles, called Remedy, and is currently stored in a MySQL database and available for export to Excel format. As part of the SciDAC Petascale Data Storage Institute (PDSI) project Collaboration this is the failure data for the High Performance Computing System-2 (MPP2) operated by the Environmental and Molecular Science Laboratory EMSL), Molecular Science Computing Facility (MSCF)[14], [42].

The MPP2 computing system has the following equipment and capabilities:

- HP/Linux Itanium-2
- 980 node/1960 Itanium-2 processors (Madison, 1.5 GHz) configured as follows:
  - ✓ 574 nodes are "fat" compute nodes with 10 Gbyte RAM and 430 Gbyte local disk
  - ✓ 366 nodes are "thin" compute nodes with 10 Gbyte RAM and 10 Gbyte local disk
  - ✓ 34 nodes are Lustre server nodes (32 OSS, 2 MDS)
  - ✓ 2 nodes are administrative nodes
  - ✓ 4 nodes are login nodes
- Quadrics QsNetII interconnect
- 11.8 TFlops peak theoretical performance
- 9.7 terabytes of RAM
- 450 terabytes of local scratch disk space
- 53 terabytes shared cluster file system, Lustre

V. METHODOLOGY

The approach we employed is to analyze the data because out of all the datasets available on Computer Failure Data Repository (CDFR) website[14][42]. The National Energy Research Scientific Computing Center (NERSC)[42] is one data that has never been analyzed or reported on in any paper. Thus, the data is examined to explore the correlations that may exist between failed hardware and the time (in years). A summary of the process is depicted in Figure 1. In this research work time is considered as the predictor variable ($\boldsymbol{X}$), hardware failures are the response variables ($\boldsymbol{Y}$), we use two machine learning algorithms, Linear Regression Model (LRM) and Support Vector Machine (SVM) with a Gaussian kernel. A linear regression model, which considers that the relationship between the predictor and response variable is linear in nature. Therefore, the relationship can be expressed as follows:

$$Y = \beta X + \alpha \tag{1}$$

Where β is the vector of regression coefficients and α is the intercept also known as an offset. The support Vector Machine model with a Gaussian Kernel assumes that the relationship between X and Y is Nonlinear in nature and can be expressed as:

$$Y = w\varphi(X) + b \tag{2}$$

Where $\varphi(X)$ is the nonlinear mapping of X using a Gaussian kernel, function given in Equation (3).

$$\varphi(X) = \varphi(\mathrm{X}) = e^{\left[-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right]} \tag{3}$$

Where $\boldsymbol{w}$ are the weights while $\boldsymbol{b}$ is the intercept.

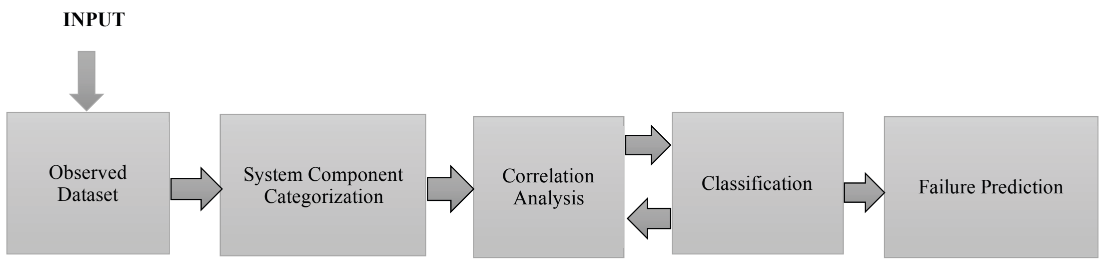


Figure 1. Failure Prediction Process

VI. EXPERIMENT AND DISCUSSION OF RESULT

Contrary to what is written on the CDFR website, the dataset covers a whole range from 2001-2006[42]. The actually data when downloaded covers from 2006 to 2008 only[42]. This data is first analyzed in this paper. The System components are categorized into seven (7) groups; Disk, DIMM (dual in-line memory module), OS, Platform, HSV, CPU, and Others. We present our obtained results using bar charts as shown in Figure 2, This enabled us have a deeper insight and better understanding of the data as well as visualizing the relationship between the individual component failures and the time (in years).

From the obtained predicted result presented in Table 1, 2 and 3, using there (3) different machine learning algorithm, it is evident that there exist a correlation between component failures and time as shown in Figure 1. In terms of the CPU failure as presented in Figure 2(a), it was observed that the number of failure in the year 2006 was over 90 while in 2007 the failure significantly decreased to about 40, and in 2008 a slight decrease was noticed above 20.

TABLE 1

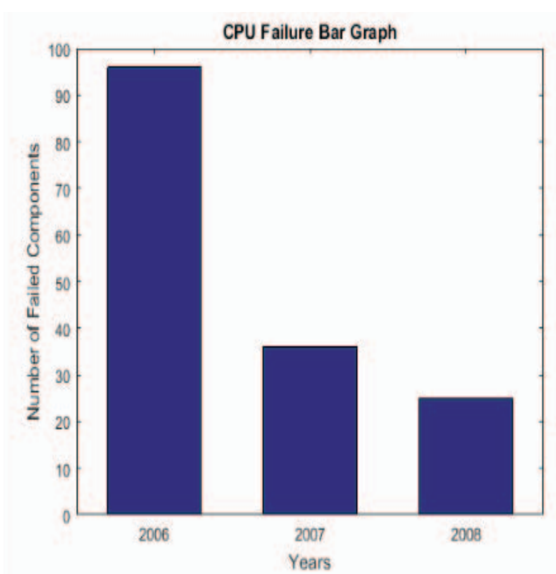
DISK			
X	linearP	GausianP	PolynomialP
2009	223	253	0
2013	0	215	0
2016	0	145	0

TABLE 2

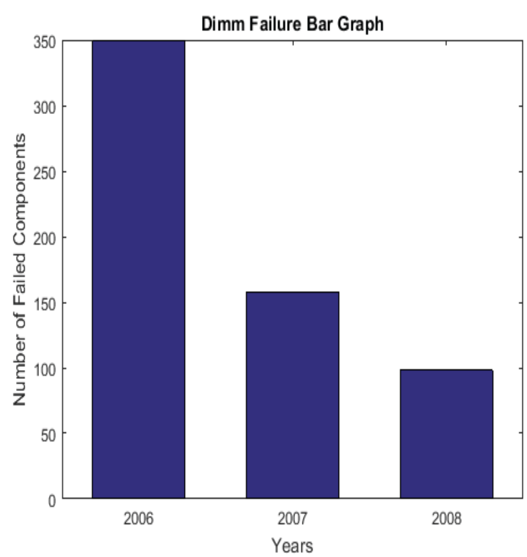
DIMM			
X	linearP	GausianP	PolynomialP
2009	88	90	0
2013	0	72	0
2016	0	23	0

TABLE 3

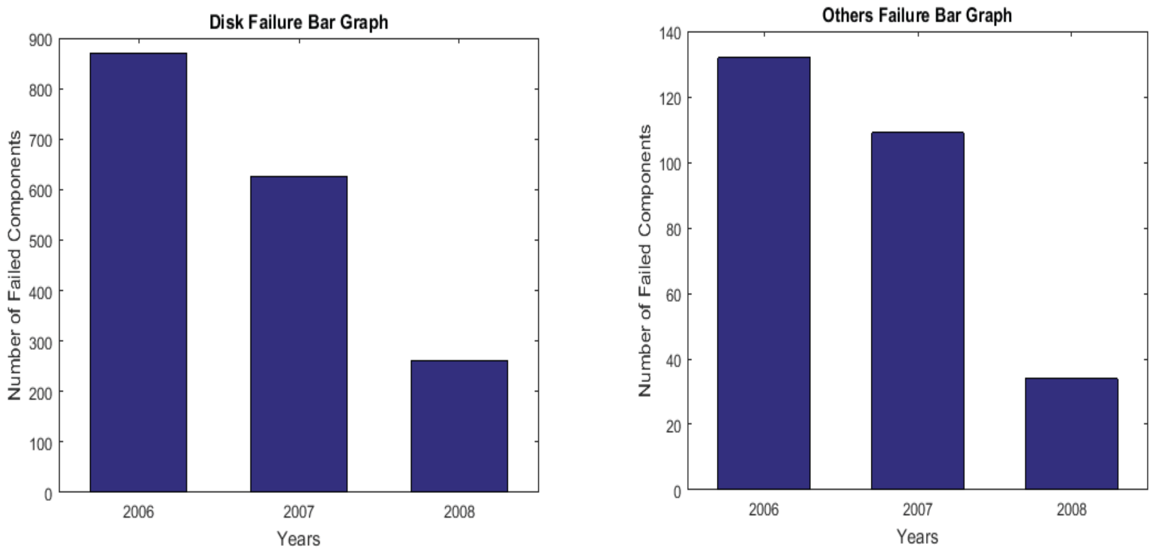
CPU			
X	LinearP	GausianP	PolynomialP
2009	18	23	0
2013	0	17	0
2016	0	10	0



(a)



(b)



(c) (d)

Figure 2. Component Failure Analysis

In another related scenario, the DIMM failure graph as presented in 2(b) indicates that in 2006 the failure was about 350, while in 2007 there was a rapid drop to about 150 and finally in 2008 it decreased to about 100.

The Disk failure graph was presented in Figure 2(c) where it was observed that in 2006 the failure increased to over 800, and in 2007 it drooped a little bit over 600 and finally in 2008, it dropped to almost 300.The results for other failures were also presented in Figure 2(d) where it was observed that in 2006 the failure was over 120, while in 2007 it went down to a little bit above 100, and finally in 2008 a large decrease was noticed where it went further down to about 30. Unfortunately, the data is insufficient for this present task. We have successfully shown the possibility of predicting some components that will fail in the future. However, as the number of predicted year's increases, both models (especially the linear model) fails, as shown in Table 1, 2, 3, thus consistently given a zero value which might be because of insufficient data obtained. We believe that better results may be achieved if the data collected spanned over 20 years. Nonetheless, we have seen that as the years come near our present day, the failure rates decrease. This can be attributed to improvement in technology, more awareness, and training and the availability of some improved fault tolerance systems.

VII. CONCLUSION

As failure becomes more prevalent in cloud systems, the ability to predict them is becoming critical. A good failure prediction model should not only focus on accuracy but also focus on how easily the obtained predicted result can be interpreted to a better fault tolerance. In this paper, we demonstrate how public available data can be invaluable regardless of the data size even though more data would have allowed us more system design insight into the data. We present an approach to failure prediction in a cloud-based environment to increase system availability using Liner Regression (LG) and Support vector machine (SVM) model respectively. In the future, we will try to present a predictive study of failure in a large real time cloud infrastructure such as the Google cluster usage trace data.

ACKNOWLEDGMENT

We would like to thank Bill Kramer and Akbar Mokhtarani from NERSC for collecting the data and sharing it.

REFERENCES

[1] R. Ghosh, L. Francesco, F. Frattini, S. Russo, and S. T. Kishor, "Scalable analytics for IaaS cloud availability," IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 57–70, 2014.

[2] T. Chalermarrewong, T. Achalakul, and S. C. W. See, "The Design of a Fault Management Framework for Cloud," 2012 9th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol., pp. 1–4, 2012.

[3] D. Sun, G. Chang, C. Miao, and X. Wang, "Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments," J. Supercomput., vol. 66, no. 1, pp. 193–228, 2013.

[4] A. Sirbu and O. Babaoglu, "Towards Data-Driven Autonomics in Data Centers," Proc. - 2015 Int. Conf. Cloud Auton. Comput. ICCAC 2015, pp. 45–56, 2015.

[5] D. Pop, "Machine Learning and Cloud Computing: Survey of Distributed and SaaS Solutions," Inst. e-Austria Timisoara, Tech. Rep 1, 2012.

[6] A. Abu-Samah, M. K. Shahzad, E. Zamai, and A. Ben Said, "Failure prediction methodology for improved proactive maintenance using Bayesian approach," IFAC Proc. Vol., vol. 48, no. 21, pp. 844–851, 2015.

[7] A. Elzamly, B. Hussin, A. Samad, H. Basari, and C. Technology, “Classification of Critical Cloud Computing Security Issues for Banking Organizations: A cloud Delphi Study,” Int. J. Grid Distrib. Comput., vol. 9, no. 8, pp. 137–158, 2016.

[8] B. Mohammed, M. Kiran, K. M. Maiyama, M. M. Kamala, and I.-U. Awan, “Failover strategy for fault tolerance in cloud computing environment,” Softw. Pract. Exp., 2017.

[9] K. Bilal, O. Khalid, S. U. . Malik, M. U. S. Khan, S. . Khan, and A. . Zomaya, “Fault Tolerance in the Cloud,” “Fault Tolerance in the Cloud” Encyclopedia on Cloud Computing. John Wiley & Sons, Hoboken, NJ, USA, 2015, pp. 291–300, 2015.

[10] ITProPortal, “ITProPortal.com: 24/7 Tech Commentary & Analysis,” 2012. [Online]. Available: http://www.itproportal.com/. [Accessed: 24-Jun-2015].

[11] D. Sheng and C. Franck, “GloudSim: Google trace based cloud simulator with virtual machines,” *Softw. - Pract. Exp.*, vol. 39, no. 7, pp. 701–736, 2015.

[12] F. Salfner, M. Lenk, and M. Malek, “A survey of online failure prediction methods,” *ACM Comput. Surv.*, vol. 42, no. 3, pp. 1–42, 2010.

[13] B. Schroeder and G. a. Gibson, “Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you,” *Conf. File Storage Technol.*, pp. 1–16, 2007.

[14] B. Schroeder and G. Gibson, “The Computer Failure Data Repository (CFDR): collecting, sharing and analyzing failure data,” *SC ’06 Proc. 2006 ACM/IEEE Conf. Supercomput.*, no. March, p. 154, 2006.

[15] B. Schroeder and G. a Gibson, “A Large-Scale Study of Failures in High-Performance Computing Systems,” *IEEE Trans. Dependable Secur. Comput.*, vol. 7, no. 4, pp. 337–350, 2010.

[16] R. K. Sahoo, M. S. Squillante, A. Sivasubramaniam, and Y. Z. Y. Zhang, “Failure data analysis of a large-scale heterogeneous server environment,” *Int. Conf. Dependable Syst. Networks, 2004*, pp. 1–10, 2004.

[17] K. Singh, S. Smallen, S. Tilak, and L. Saul, “Failure analysis and prediction for the CIPRES science gateway Kritika,” *Concurr. Comput. Pract. Exp.*, vol. 22, no. 6, pp. 685–701, 2016.

[18] G. H. Thomas Gentner, Klau p. Gungl, “Patent US9319030 - Integrated circuit failure prediction using clock duty cycle recording and,” 2016.

[19] A. Khan, B. Bussone, J. Richards, and A. Miguel, “A practical Approach to Hard Disk Failure Prediction in Cloud Platforms,” in *2016 IEEE Second International Conference on Big Data Computing Service and Applications ??*, 2016, pp. 105–116.

[20] T. Samak, D. Gunter, M. Goode, E. Deelman, G. Juve, F. Silva, and K. Vahi, “Failure analysis of distributed scientific workflows executing in the cloud,” *Proc. 2012 8th Int. Conf. Netw. Serv. Manag. CNSM 2012*, pp. 46–54, 2012.

[21] Rajashekarappa and K. M. Sunjiv Soyjaudah, “Self Monitoring Analysis and Reporting Technology (SMART) Copyback,” *Commun. Comput. Inf. Sci.*, vol. 157 CCIS, pp. 463–469, 2011.

[22] S. A. E. Keke Gai, Meikang Qiu, “Security-Aware Information Classifications Using Supervised Learning for Cloud-Based Cyber Risk Management in Financial Big Data,” in *2016 IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security*, 2016, pp. 197–202.

[23] L. Zhang, K. Rao, R. Wang, and Y. Jia, “Risk Prediction Model Based on Improved AdaBoost Method for Cloud Users,” *Open Cybern. Syst. Journal, 2015*, vol. 9, pp. 44–49, 2015.

[24] S. Büsch, V. Nissen, and A. Wünscher, “Automatic classification of data-warehouse-data for information lifecycle management using machine learning techniques,” *Inf. Syst. Front.*, 2016.

[25] D. Fall, T. Okuda, Y. Kadobayashi, and S. Yamaguchi, “Risk Adaptive Authorization Mechanism (RAdAM) for Cloud Computing,” *J. Inf. Process.*, vol. 24, no. 2, pp. 371–380, 2016.

[26] C. Guo, Y. Liu, and M. Huang, “Obtaining Evidence Model of an Expert System Based on Machine Learning in Cloud Environment,” *J. Internet Technol.*, vol. 16, no. 7, pp. 1339–1349, 2015.

[27] Z. Amin, N. Sethi, and H. Singh, “Review on fault tolerance techniques in cloud computing,” *Int. J. Comput. Appl.*, vol. 116, no. 18, pp. 11–17, 2015.

[28] A. Pellegrini, P. Di Sanzo, and D. R. Avresky, “Proactive Cloud Management for Highly Heterogeneous Multi-cloud Infrastructures,” in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2016, pp. 1311–1318.

[29] S. P. P. K.S. Thakur., T. R.Godavarthi., “10.1.1.416.6042,” vol. 3, no. 6, pp. 698–703, 2013.

[30] A. Bellet, A. Habrard, and M. Sebban, “A Survey on Metric Learning for Feature Vectors and Structured Data,” 2013.

[31] P. Mell, T. Grance, and T. Grance, “The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology,” *Natl. Inst. Stand. Technol. Spec. Publ. 800-145 7 pages*, 2011.

[32] M. Fu, L. Zhu, D. Sun, A. Liu, L. Bass, and Q. Lu, “Runtime recovery actions selection for sporadic operations on public cloud,” *Softw. - Pract. Exp.*, vol. 39, no. 7, pp. 701–736, 2016.

[33] D. Gnanavelu and D. G. Gunasekaran, “Survey on Security Issues and Solutions in Cloud Computing,” *Int. J. Comput. Trends Technol.*, vol. 8, no. 8, pp. 126–130, 2014.

[34] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, “DDoS attack protection in the era of cloud computing and Software-Defined Networking,” *Comput. Networks*, vol. 81, pp. 308–319, 2015.

[35] M. Amoon, “A Framework for Providing a Hybrid Fault Tolerance in Cloud Computing,” pp. 844–849, 2015.

[36] R. Jhawar, V. Piuri, and I. Universit, “Fault Tolerance Management in IaaS Clouds,” *2012 IEEE First AESS Eur. Conf. Satell. Telecommun.*, pp. 1–6, 2012.

[37] Z. Pantic and M. Babar, "Guidelines for Building a Private Cloud Infrastructure," *ITU Tech. Rep. - TR-2012-153TR-2012-153*, 2012.

[38] C. Pahl and H. Xiong, "Migration to PaaS clouds - Migration process and architectural concerns," *2013 IEEE 7th Int. Symp. Maint. Evol. Serv. Cloud-Based Syst.*, pp. 86–91, Sep. 2013.

[39] O. Sefraoui, M. Aissaoui, and M. Eleuldj, "Cloud computing migration and IT resources rationalization," *2014 Int. Conf. Multimed. Comput. Syst.*, pp. 1164–1168, Apr. 2014.

[40] A. Sen and S. Madria, "Off-Line Risk Assessment of Cloud Service Provider," *2014 IEEE World Congr. Serv.*, pp. 58–65, Jun. 2014.

[41] S. Yadav, "Comparative Study on Open Source Software for Cloud Computing Platform : Eucalyptus , Openstack and Opennebula," *Res. Inven. Int. J. Eng. Sci. Vol.3, Issue 10*, vol. 3, no. 10, pp. 51–54, 2013.

[42] B. Schroeder and G. Gibson, "The computer failure data repository (CFDR)," *... Reliab. Anal. Syst. Fail. Data ...*, no. March, p. 6, 2007.