Prediction of SLA Violation in Cloud Resource Allocation using Machine Learning based Back Propagation Neural Network (BPNN)

Karthik Kambhampati, A.Srinagesh

Abstract*: Cloud computing is defined as a pay-per-use model which would offer the user required services. Cloud computing services can be identified as three models based on the type of resources and the services they are provisioning. Cloud Service models that are more popular in cloud computing environment are, "SaaS, IaaS, and PaaS". Among these cloud service models, SaaS model provides flexible and reliable services to the cloud users based on their requirement. SaaS rents the required resources from the IaaS cloud Service providers to offer the services to the cloud users in a reliable manner. However, this rental scheme would increase the administration and maintenance costs. And also, resources that are rented from the external cloud resource providers might degrade the service quality in hosting services due to lack of SLA agreement. Several research works have been conducted previously for performing a better admission control. Some of the research works are discussed here for better understanding of the merits and demerits of existing research efforts. In this paper we are focusing on SLA violation prediction by both user as well as CSP. For to predict SLA violation we use machine learning based back propagation network model. The experimental results shows that BPNN performed well when compared to the statically models.

Index Terms*: Cloud, SLA, BPNN, cloud service provider, resource allocation.

I. INTRODUCTION

Cloud computing is defined as a pay-per-use model which would offer the user required services. Cloud computing services can be identified as three models based on the type of resources and the services they are provisioning. Cloud Service models that are more popular in cloud computing environment are, "SaaS, IaaS, and PaaS". Among these cloud service models, SaaS model provides flexible and reliable services to the cloud users based on their requirement. SaaS. SLA satisfaction is a very important factor in cloud computing environment and must be considered for better satisfaction of user requirements through successful task execution. There might be occurrence of more problems while attempting to perform admission control considering the SLA factors.

Software-as-a-Service provider is a popular service providers who maintain the software distribution model and provide the required software services to the cloud users based on their requirements. SaaS providers focus mainly on distribution of software services and they do require sufficient resources like storage for the better provisioning of them. IaaS service providers also offer storage related services to the cloud users in pay per use terms. The storage service provided by the IaaS service providers can be utilized by SaaS service providers to improve the performance of task execution. There are many cloud based real world applications that rent the resources from the IaaS service providers for completion of their tasks. SLA violation is the critical issue which might occur while renting the resources from the external service providers for internal deployment. This problem might create more impact on the software services that are allocated to cloud users in terms of increased administration and maintenance cost. This problem needs to be resolved in an effective manner for the better resource allocation to the cloud users. Hence the better admission control mechanism needs to be implemented between the SaaS providers and the IaaS providers which can control the renting resources in an effective manner by checking the QoS constraints.

The proposed research methodology of this work makes use of two machine learning algorithms for efficient admission control with the satisfaction of various QoS requirements by learning the machine status information. The proposed research attempts to improve the satisfaction level of both cloud users and cloud service providers by considering SLA parameters of both. The admission control algorithm proposed in this work will analyze the various user QoS requirement and learned knowledge of machine status and then will find the suitable resources for better allocation. The scheduling would be performed based on the admission control Algorithms decision.

A. Roles Involved

The various roles and entities that are involved in the proposed research methodology of cloud computing environment are

- Users
- SaaS providers
- IaaS providers

Each entity mentioned above performs its role in an effective manner to reach the optimal and better admission control process over the heterogeneous cloud computing environment.

Revised Manuscript Received on June 07, 2019.

Karthik Kambhampati, Research Scholar, Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur.

Dr.A.Srinagesh, Associate Professor, Department of Computer Science and Engineering, RVR&JC College of Engineering, Guntur,

B. Users

Cloud users are the ones who will request the application for execution in the cloud computing environment along with the QoS parameters. Cloud users will submit their task requests to the SaaS service providers to get access to the contents they require. On the user side of process, the requests are sent through the SaaS supplier claim along with the QoS constraints namely the economical, target and punishment rate. After that, the admittance control and algorithms for scheduling are initialized to allow or reject the request where acceptance indicates that both the parties have accepted the QoS requirements.

C. SAAS service providers

SaaS providers are the ones who allocate software services to the cloud users based on their requirement. The SaaS provider uses resources from IaaS providers and rents SaaS to the users with the aim of minimizing the functional cost using IaaS resource providers and improving Customer Satisfaction Level (CSL) by providing SLA with guaranteed QoS requirements. Between SaaS providers, users and the resource provider’s two layers of SLA are utilized. Thus SLA with resource providers can enforce resource providers to deliver satisfactory service. When the contract is violated by any user, the defaulter has to pay penalty as defined in the service agreement.

D. IAAS service providers

An IaaS supplier offers Virtual Machines to SaaS suppliers and it is in control for dispatching pictures of VM to process on their physical assets. The SaaS supplier stage layer utilizes pictures of VM to create occasions. It is important to set up SLA with an asset supplier called SLA(R), in light of the fact that it encourages the asset supplier to fulfill the QoS constraints.

Quality of Service plays vital role in the cloud computing environment where every user needs to complete their tasks as they want. In cloud computing environment, both cloud users and cloud service providers have their own requirements to fulfil objectives such as profit, reputation and so on. In the proposed research methodology, QoS constraints of both service providers and cloud users are considered in admission control. SLA agreement is defined as the agreement made between cloud service providers and cloud users in terms of sets of constraints. SLA agreement would be made, once both the cloud users and the cloud service providers agree with each other based on their QoS constraints. This SLA agreement should be maintained until completion of all the tasks that are submitted by the users. QoS constraints considered for SLA agreement which is made in terms of both cloud service provider and the cloud user for achieving better admission control is given in the following sub sections.

E. Service Level Agreement Constraints of Cloud Users

The algorithms employed to allow or scrap a request. A formal agreement-SLA (U) is signed between the user and provider only when the request is accepted and the QoS requirements can be guaranteed with the subsequent things:

Target: Greatest time user waits for the result.

Budget: sum willing to pay for the wished services.

Penalty Rate Ratio: Sum given for consumers pay when the SaaS supplier misses the due date.

Input File Size: The size of users input file.

Request Length: Measure of Millions of Instructions (MI) obligatory to be implemented to serve the specific users demand

F. Service Level Agreement Constraints of Cloud Service Providers

An IaaS provider offers Virtual Machines to SaaS providers and the images of VM are dispatched to execute on the physical machines. SaaS provider utilizes the VM images to generate the instances. The establishment of SLA with SLA (R) is necessary as it enforces the resource provider to assure QoS and also provides jeopardy of transfer for SaaS providers if SLA is violated by the resource provider. The SLA(R) includes the following properties:

Service Initiation Time: Time taken to deploy a VM.

Price: Amount the SaaS provider has to pay per hour for using a VM from a resource provider

Input Data Transfer Price: Amount the SaaS provider has to pay for data transfer from local machine (their own machine) to resource provider’s VM.

Output Data Transfer Price: Sum the SaaS supplier needs to pay for information exchange from asset provider‟s VM to local machine.

Processing Speed: Speed at which VM is processing. Million Instructions per Second (MIPS) is used as a unit of a VM‟s handling speed.

Data Transfer Speed: The speed at which the information is exchanged. It depends on the area remove and furthermore the system execution.

The rest of the paper is organized as follows second section describes existing literature, setion-3 explains proposed mechanism, preceding section describes experimental execution and last section concludes the paper.

II. RELATED WORK

Resource allocation is the most complex process in the cloud computing environment in which various factors need to be considered for achieving the required user satisfaction levels. Catak & Balaban (2012) introduced the methodology for allocation resources in an optimized and accurate manner for the cloud consumers by learning the resources status. This learning process is accomplished by using the SVM approach which will allocate resources in the optimal method by adjusting the fitness curve values. This approach can provide accurate results with an improved user satisfaction

level. Lee et al. (2010) developed a methodology called the SLA aware resource allocation which would allocate resources that can satisfy the service level agreement requirements mentioned by the users. Reig et al. (2010) attempted to allocate resources with consideration of the Quality of Service (QoS) parameter called the deadline. This approach can select resources in the SaaS service provider environment which can complete the user submitted tasks based on the SLA constraints specified by the user. Linlin Wu et al. (2012) introduced an SLA based admission control mechanism.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

This work attempts to allocate resources optimally in the SaaS service provider's Environment considering the user satisfaction level and profit of the SaaS service provider.

Mardente et al. (2004) considered to make decisions for accepting or rejecting Service Level Agreements (SLAs) requested by a user which is known as admission control. The admission control process is used to ensure the availability of provisions below the SLA. Also the work addresses the problem of the SLA optimal set of paths along which traffic might be low.

Dimokritos Stamatakis & Olga Papaemmanouil (2014) proposed a methodology to dole out approaching inquiry preparing outstanding burdens to the held assets with the goal that SLA infringement are decreased. This work likewise talked about the plan of a structure that empowers the particular of custom application level execution SLAs and offers remaining task at hand administration systems that can consequently tweak their usefulness towards gathering the application-explicit SLAs.

Rahul Garg et al. (2009) introduced a SLA based structure for QoS provisioning and dynamic limit portion. The proposed SLA enables clients to purchase a long haul limit at a pre-determined cost. Be that as it may, the client may progressively change the limit assignment dependent on the immediate interest. This work additionally manages a three level estimating model with punishments SLA that offers motivating forces to the clients to surrender unused limits and procure greater limit as required. This work is considered as a down to business initial move towards an increasingly unique estimating situation.

Jian Pu et al. (2005) designed a framework to perform reliable concerned SLA guaranteed admission control over the resource allocation environment. This framework will gather the SLA requirements from the user based on which initial resource allocation would be done.

III. Proposed Work

Here we presented a model for classification of SLA violation in cloud resource allocation form the perspective of user and CSP way. For that here we use data set WS-DREAM and we take 5 sub data sets and made analysis of that data. After words we use sampling method for training and testing of the model. For that we use 80% data as training and given it as input for training of proposed BPNN model. Remaining 20% would be used as testing data for validating the model performance.

A. BPNN Algorithm

The BPNN calculation contains 3 layers, for example, information, yield and concealed layer. The BPNN calculation is accustomed to ascertaining the mistakes of the yield layer to discover the blunders in the shrouded layers. The angle plunge technique used to figure the loads and changes made to the system to limit the yield blunder. The BPNN calculation has turned into the standard calculation utilized for preparing multilayer recognition. In the first place, discover the blunders between the real and the ideal yields.

$$E_p = \sum_{i=1}^{j}(e_i)^2 \qquad (1)$$

Where,

e_i is error signal.

P signifies in the p^th^ pattern;

j is the integer of the output units.

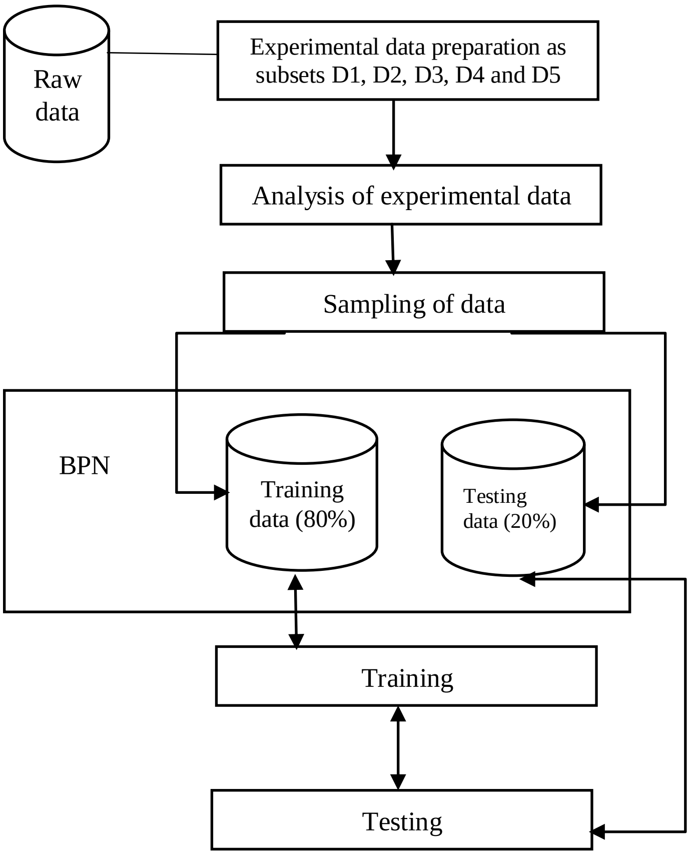

Raw data
Experimental data preparation as subsets D1, D2, D3, D4 and D5
Analysis of experimental data
Sampling of data
BPN
Training data (80%)
Testing data (20%)
Training
Testing

Figure. 1 proposed model architecture

The gradient descent method given by

$$W_{Ki} = -\mu \frac{\partial E_P}{\partial W_{Ki}} \qquad (2)$$

The BP computes miscalculations in the output phase ∂_i, and the hidden phase ∂_j, are using the formulas

$$\partial_i = \mu(d_i - y_i)f^1(x_i) \qquad (3)$$

$$\partial_j = \sum_i \partial_i w_{ij} f^1(y_i) \qquad (4)$$

The fault obtained during back propagation phase is used to update the weights and biases in both the hidden and output layers. The weights, w_{ij} and biases, b_i, then adjusted using the following formulas:

$$w_{ij}(k+1) = w_{ij}(k) + \mu\partial_j y_i \qquad (5)$$

$$w_{ij}(k+1) = w_{ij}(k) + \mu\partial_j x_i \qquad (6)$$

$$b_j(k+1) = b_i(k) + \mu\partial_j \qquad (7)$$

k is the number of the epoch and μ is the learning rate.

B. BPNN Algorithm Steps

- *Arbitrarily pick the underlying loads.*
- *While mistake is excessively huge;*

For each preparation design (introduced in arbitrary request)

Apply the contributions to the system.

Compute the yield for each neuron from the information layer, through the concealed layer(s), to the yield layer.

Ascertain the blunder at the yields.

Utilize the yield mistake to register blunder signals for pre-yield layers.

Utilize the mistake sign to figure weight modifications.

Apply the weight alterations.

- *Periodically evaluate the framework execution.*

Apply Inputs from a Pattern

- *Apply the estimation of each info parameter to each information hub*
- *Input hubs figure just the character work*

Figure Outputs for Each Neuron Based On the Pattern

- *The yield from neuron j for example p is* o_{pj} *where*

$$o_{pj}(net_j) = \frac{1}{1+e^{-\tau net_j}} \tag{8}$$

k varieties over the inputs and Wjk is the weight on the connection from input k to neuron j

Compute the Error for Each Output node

- *The output neuron error signal* d_{pj} *is given by*

$$d_{pj} = (T_{pj} - O_{pj})O_{pj}(1 - O_{pj})$$

Compute the Error Signal for Each Hidden Node

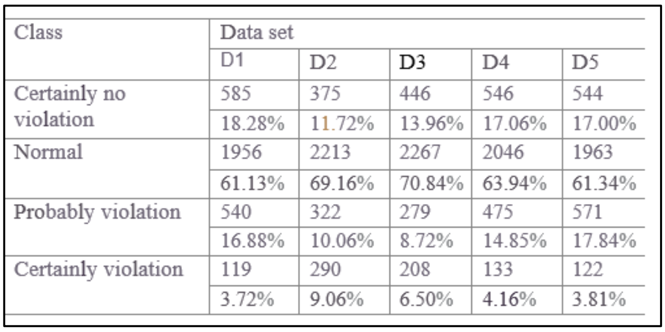
- *The hidden neuron error signal* δ_{pj} *is given by.*

$$\delta_{pj} = O_{pj}(1 - O_{pj}) \sum_k \delta_{pk} W_{kj} \tag{9}$$

IV. Experimental Analysis

Cloud QoS infringement distinguishing pieces of proof utilizing AI based BPNN. Here we utilize WS-DREAM [4] informational collection in our analyses to decide an ideal multiclass order model for recognition and expectation of cloud QoS infringement. WS-DREAM [4] informational collection as created by research in [21] contains hints of genuine world QoS assessment results from 142 clients on 4,532 Web Services more than 64 distinctive availabilities. WS-DREAM informational collection reflects Web Service in administration processing and distributed computing. Each occasion of WS-DREAM comprises of three highlights, for example, the SLO measurements of reaction time in a flash, throughput in kilobytes every second (kbps), and Time Slot ID speaking to the 64 distinctive schedule openings in which each vacancy serves a period zone with 15-min time interim. Reaction time estimates the time interim between when a solicitation is made by client and when the reaction is gotten by the customer, while throughput computes the measure of remaining burden per unit of time over the estimation time frame [22]. For our tests, we chose from WS-DREAM [4] informational index, 50 clients on 5 distinct servers on 64 diverse vacancies and structure into five sub datasets, in particular informational collection D1, D2, D3, D4 and D5. Every one of these sub datasets contains an aggregate of 3200 quantifiable estimations of reaction time with comparing throughput esteems. Since we are actualizing directed learning method, all information will be named with joint choice guidelines which will be talked about at resulting passages of the paper. These five sub informational collections are again isolated into two bits, one with 80% of the information as preparing dataset and another with 20% of the information as testing informational index, testing for model characterization and expectation precision.

Table. 1 various violation levels identified by proposed BPNN



Class	Data set				
	D1	D2	D3	D4	D5
Certainly no violation	585	375	446	546	544
	18.28%	11.72%	13.96%	17.06%	17.00%
Normal	1956	2213	2267	2046	1963
	61.13%	69.16%	70.84%	63.94%	61.34%
Probably violation	540	322	279	475	571
	16.88%	10.06%	8.72%	14.85%	17.84%
Certainly violation	119	290	208	133	122
	3.72%	9.06%	6.50%	4.16%	3.81%

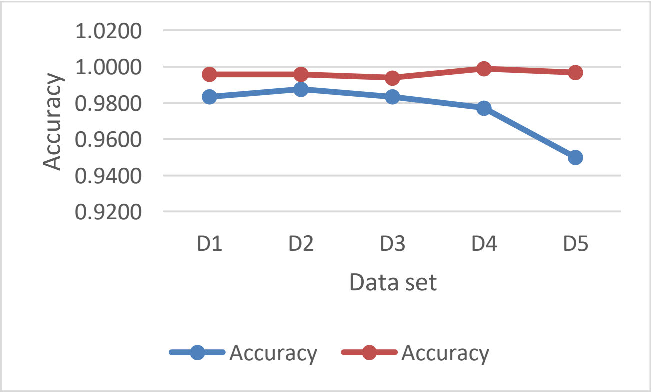


Figure. 2 Accuracy

Accuracy - Accuracy is the most inherent exhibition extent and it is essentially a fraction of accurately a waited view to the absolute perceptions.

Accuracy = TP+TN/TP+FP+FN+TN

Here figure 2 shows the accuracy comparison of proposed BPNN and existing model with respect to various data sets we considered for experimental purpose. Figure shows that the four different data values proposed mechanism shows good accuracy.

Recall is the proportion of accurately awaited positive perceptions to the all perceptions in real class - yes.

Recall = TP/TP+FN

Here figure 3 shows the recall comparison of proposed BPNN and existing model with respect to various data sets we considered for experimental purpose. Figure shows that the four different data values proposed mechanism shows good recall.

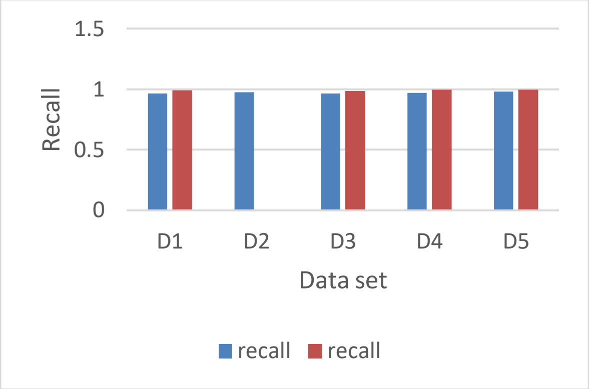


Figure. 3 Recall

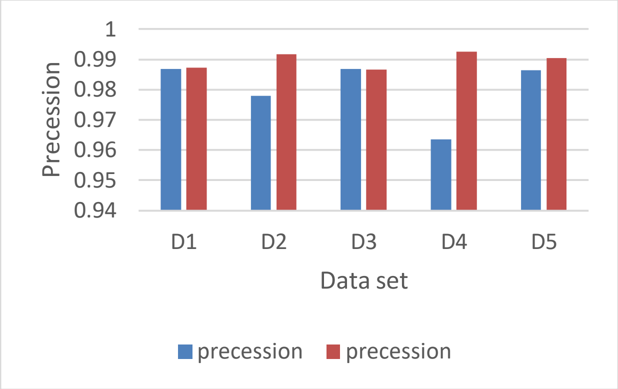


Figure. 4 Precession

Precision is the fraction of exactly predicted affirmative perceptions to the absolute estimated positive perceptions.

Precision = TP/TP+FP

Here figure 4 shows the Precision comparison of proposed BPNN and existing model with respect to various data sets we considered for experimental purpose. Figure shows that the four different data values proposed mechanism shows good Precision.

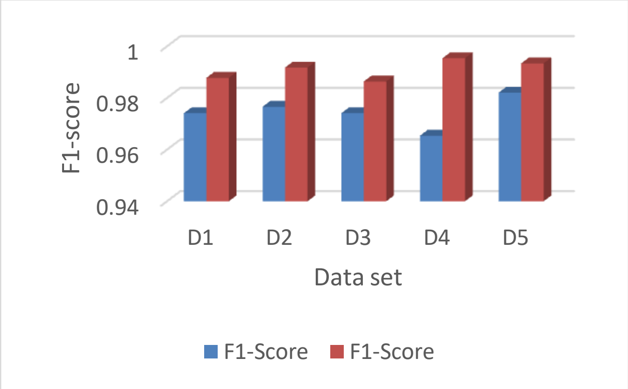


Figure. 5 F1-Score

F1 Score is the weighted normal of Precision and Recall. In this manner, this score considers both false +ve and false –ve.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Here figure 5 shows the F1 Score comparison of proposed BPNN and existing model with respect to various data sets we considered for experimental purpose. Figure shows that the four different data values proposed mechanism shows good F1 Score.

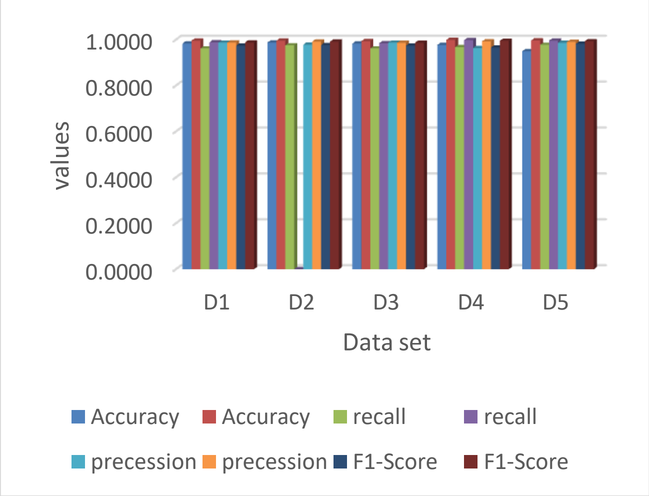


Figure. 6 model performance

Here figure 6 shows the overall outcome comparison of proposed BPNN and existing model with respect to various data sets we considered for experimental purpose.

V. CONCLUSION

Cloud Service Provider to foresee the event of the infringement of administrations dependent on response time and throughput. At the point when there is a propensity for the exchanges to go past as far as possible, framework executive will take vital preventive measures to take the framework back to typical conditions. This will decrease the opportunity for infringement to happen, henceforth moderating lose or exorbitant punishment. All things considered, if infringement does happen, therapeutic activity must be set up to anticipate future event. The proposed method identifies the SLA violation in more accurate manner. The performance results show the accuracy of identifying the SLA violation.

REFERENCES

1. GMell, P., Grance, T.: The NIST definition of cloud computing. national institute of standardsand technology. U.S. Department of Commerce, Special Publication 800-145 (2011)
2. Mirobi, G.J., Arockiam, L.: Service level agreement in cloud computing: an overview. In:International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT, pp. 753–758. IEEE, Kumaracoil (2015). https://doi.org/10.1109/iccicct.2015.7475380
3. OSG Cloud Working Group: Report on Cloud Computing to the OSG Steering Committee. https://www.spec.org/osgcloud/docs/osgcloudwgreport20120410.pdf. Accessed 20 July 2017
4. WSDREAM Data Set. https://github.com/wsdream/ wsdream-dataset/tree/master /dataset2.Accessed 20 July 2017
5. R version 3.4.3: A language and environment for statistical computing. https://wbc.upm.edu.my/cran/. Accessed 20 July 2017
6. Emeakaroha, V.C., Ferreto, T.C., Netto, M.A.S., Brandic, I., De Rose, C.A.F.: CASViD:application level monitoring for SLA violation detection in clouds. In: IEEE 36th AnnualComputer Software and Applications Conference. IEEE, Izmir (2012). https://doi.org/10.1109/compsac.2012.68

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

7. Musa, S.M., Yousif, A., Bashi, M.B.: SLA violation detection mechanism for cloudcomputing. Int. J. Comput. Appl. 133, 8–11 (2016)
8. Leitner, P., Wetzstein, B., Rosenberg, F., Michlmayr, A., Dustdar, S., Leymann, F.: Runtime prediction of service level agreement violations for composite services. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave -2009. LNCS, vol. 6275, pp. 176–186. Springer,Heidelberg (2010). https://doi.org/10.1007/978-3-642-16132-2_17
9. Hani, A.F.M., Paputungan, I.V., Hassan, M.F.: Support vector regression for service levelagreement violation prediction. In: International Conference on Computer, Control,Informatics and its Applications, IC3INA. IEEE, Jakarta (2013)
10. Tang, B., Tang, M.: Bayesian model-based prediction of service level agreement violations for cloud services. In: Theoretical Aspects of Software Engineering Conference, TASE.IEEE, Changsha (2014)
11. Sheng, D., Kondo, D., Cirne, W.: Host load prediction in a Google compute cloud with aBayesian model. In: Proceedings of the International Conference on High PerformanceComputing, Networking, Storage and Analysis, SC 2012. IEEE, Salt Lake City (2012)
12. Hemmat, R.A., Abdelhakim, H.: SLA violation prediction in cloud computing: a machinelearning perspective. eprint arXiv:1611.10338 (2016)
13. Smola, A., Vishwanathan, S.V.N.: Introduction to Machine Learning, 1st edn. Cambridge University Press, Cambridge (2008) 512 T.-S. Wong et al.
14. Chang, C.C., Lin, C.J.: LIBSVM: a library of support vector machine. ACM Trans. Intell.Syst. Technol. 2, 1–27 (2011)
15. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. 13, 415–425 (2002)
16. Rish, I.: An empirical study of the naive Bayes classifier. In: IJCAI 2001 Workshop onEmpirical Methods in Artificial Intelligence, vol. 3, pp. 41–46 (2011)
17. Popescu, M.C., Balas, V.E., Perescu-Popescu, L., Mastorakis, N.: Multilayer perceptron andneural networks. WSEAS Trans. Circ. Syst. 8, 579–588 (2009)
18. Bishop, C.M.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
19. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996.Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
20. Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
21. Zheng, Z., Zhang, Y., Lyu, M.R.: Investigating QoS of real-world web services. IEEE Trans. Serv. Comput. 7, 29–32 (2014)
22. IBM Informix Documentation Team: IBM Informix Performance Guide. Version 12.10.IBM, USA (2016)
23. Hossin, M., Sulaiman, M.N.: A review on evaluation metrics for data classificationevaluations. Int. J. Data Min. Knowl. Manag. Process (IJDKP) 5(2), 1–11 (2015). https://doi.org/10.5121/ijdkp.2015.5201
24. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. 27, 861–874 (2006)
25. McHugh, M.L.: Interrater reliability: the kappa statistic. Biochem. Med. 22(3), 276–282 (2012)

AUTHORS PROFILE

Karthik Kambhampati pursued his B.Tech Degree in Electronics and Communiation Engineering from JNTU,Kakinada in 2011 and obtained his M.Tech Degree from Andhra University, Visakhapatnam in Distinction in 2013. He has worked for Computer Science Corporation during 2013 to 2016. He has been pursuing Ph.D Degree in Computer Science Engineering from Acharya Nagarjuna University, India. Presently he is working as Technical Lead in Salesforce Services, Tampa,FL.