2018 International Conference on Advances in Computing and Communication Engineering (ICACCE-2018)
Paris, France 22-23 June 2018

K-mean Clustering in Web Service Quality Datasets Using AWS and RapidMiner

Kriti Gupta
ASET, Department of CSE,
Amity University Uttar Pradesh
India
kritig4@gmail.com

Arjun Vaibhav Srivastava
ASET, Department of CSE,
Amity University Uttar Pradesh
India
arjunvaibhavspringer@gmail.com

Gaurav Raj
ASET, Department of CSE,
Amity University Uttar Pradesh
India
graj@amity.edu

***Abstract*— Today, number of research works are under process using computing resources delivered as a service over the network. This paper focuses in the field of cloud computing and proposes the model for service clustering of cab services using k-means clustering. This paper proposes a highly competent and efficient methodology. Our research includes the web service development using AWS, real time data collection and its analysis based on clustering. The web services are designed to provide approximate time and distance as per client inputs and then produces the output in terms of cost of different cab services provided by the cab service providers. Through our clustering approach, we are proposing the better approach for selecting the best service using data analysis of web service quality dataset over AWS based developed model.**

Keywords—cloud computing, AWS, web service, prediction, real time datasets

I. INTRODUCTION

The term 'cloud computing' originated in 1996. This technology was boosted by amazon.com when it released its product elastic cloud computing (EC2) and Amazon Web Service (AWS) . Since then, this technology has demand in almost every sector of IT industry, especially in management and analysis of on demand resources.

Cloud computing is a practice of deploying several virtual machines on the Internet so as to store, manage, analyze and process data anywhere around the world. It performs the delivery of pay per use of computing resources like, networking, database, software, storage, analytics, and much more resources online.

Cloud computing provides three types of services:

1. *IaaS (Infrastructure as a Services):* Vendors use this service to provide users with resources related to computing like servers, network and storage.

2. *SaaS (Software as a Service):* This makes use of internet in order to provide application to users that are being controlled by third party.

3. PaaS (Platform as a Service): A structure is provided to the developers in which they can build and use it to create.

These services help in increasing elasticity, scalability and availability, load balancing in cloud computing.

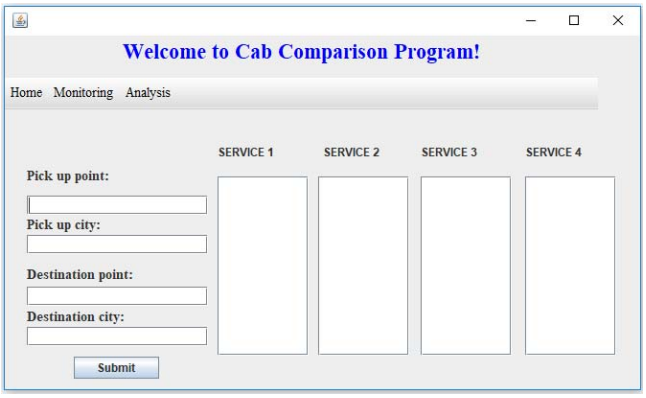

Fig. 1. GUI Framework of the proposed model

The purpose of this paper is to deploy a web service in AWS and analyze the real-time results. The proposed mathematical model helps in predicting the best cab service to the customer based on its availability, cost and response time. The cost estimation calculators of four Indian based cab services as SERVICE 1, 2, 3 and 4 are deployed on the AWS Elastic Beanstalk machine. The results are much precise and accurate as compared to existing methodologies.

The research paper is organized as follows: Section II consists of a detailed analysis of existing methodologies. Section III consists of the proposed methodology of this paper followed by Section IV which shows the result of this methodology and Section V gives the conclusion to this research work.

II. LITERATURE REVIEW

As data is increasing rapidly the necessity to manage and analyze such large volume of data is also increasing. Cloud computing has been observed as an ultimate approach to solve the problem of analyzing the increasing data globally. Kumbhare et al. [1] have proposed a model which displays the concept of "dynamic dataflows". They have proposed two greedy heuristic algorithm: shared and centralized which is constructed on algorithm based on variable sized bin packing & then do compare it with the Genetic Algorithm(GA) which gives a near optimal solution. Thus, by alternate

978-1-5386-4485-0/18/$31.00 ©2018 IEEE

implementation of dataflow tasks users have choice in areas of service composition and an add new method to overcome the issues of the services and applications. [2] He et al. have adopted the Linear Temporal Logic (LTL) to consider the multiple competing challenges faced at datacenters. Their paper deals with job scheduling of two hundred MapReduce jobs on the Amazon Web Service. The results prove that their method is much better in balancing the multiple conflicting objectives as compared to the traditional methods.

Yassin et al. discuss about the security issues faced by the users of cloud [3]. As a client uses pay as you go service provided by the SaaS (Software as a Service) provider, the user must give their information to the internet which makes them vulnerable to threat. One such threat presented in their paper is about SQL Injection which paves way for the attackers to violate the confidentiality, availability and vulnerability of the cloud users. Therefore, the authors proposed a detection framework solution to resolve the security issues related to SaaS customers.

Resource scheduling is among one of the important problems of cloud computing. [4] Lee et al. have presented DeepSpotCloud in their paper. It is implemented with the help of AWS cloud computing services in order to serve tasks of real deep learning. Their proposed model of Billing Policy – Hourly migration achieves 13% more gain in cost as compared to interrupt driven scheduling policy. Another scheduling method that is PLAStiCC: Predictive Look-Ahead Scheduling [5] has been proposed by Kumbhare et al. They have proposed a solution to the excessive stream of continuous flowing data through scheduling strategies. With the help of PLAStiCC methodology which is a predictive based lookahead model they could show an refinement of 20% in the overall profit in comparison to algorithm of reactive adaptation.

[6] Zinno et al. have proposed a solution to increase the efficiency of P-SBAS algorithm which is extended version of DInSAR technique. The results prove that with the help of cloud computing and appropriate scheduling of parallel jobs they were able to generate results of DInSAR on a large scale of 150,000 km2 in a very short span of time of about 9 hours. Another research carried out by Khodadadi et al. [7] shows an IoT application using cloud computing. In their paper they have proposed a framework that is data-centric though which IoT applications such as sensors can communicate with each other with the help of a prototype which is executed on AWS that is constructed on the top of the Platform of Aneka Cloud Application.

For the purpose of managing virtual machines (VMs) in the Amazon Web Services EC2 public cloud Grimaldi et al. have proposed a feedback mechanism [8] in which the policy of gain scheduling is evaluated with different workloads and it is compared to the robustness of algorithm at the time of VM failures. Thus, the results show high performance at the time of both constant and time-varying workloads. Arabnejad et al. proposed a Deadline Distribution Ratio (DDR) scheduling algorithm [9] so as to minimize the cost across majority of deadlines along with sustaining a high scheduling success-rate. The results show that their methodology has successful outcome in yielding the lowest cost. CloudAnalyst [10] is a tool extended over CloudSim which is helping in analysis and modelling of advance computing environments. Wickremasinghe et al. have proposed CloudAnalyst in their paper and have discussed the various features of the simulator. It can implement and showing results of large scale cloud computing environment but is incapable of taking real-time data centers as inputs. Therefore, this paper proposes an algorithm that is capable of taking input of real-time data sets and then implementing task scheduling on it.

III. PROPOSED METHODOLOGY

This paper proposes an algorithm which is capable of taking real time data sets as input from web services developed in AWS and processing it to give an analysis of the data through a simulator. AWS provides the platform to host website and private cloud on virtual machines with the help of instances. Services in AWS includes Elastic Compute Cloud(EC2) and Simple Storage Service(S3). For the same, firstly an account is made in Amazon Web Services and then EC2 instance is launched. The following steps are performed in the deploying of web services and predicting the best cab service based on the obtained results.

Algorithm: To obtain real time datasets from AWS

***1**: Take Input from the user about source and destination.*
***2:** Calculate estimated distance and time from Google API*
***3:** Select Elastic Beanstalk service from AWS console*
***4:** Select create new application*
***5:** Enter application name and description*
***6:** Select 'create web server' option.*
***7:** Select 'Tomcat' in predefined configuration and click on next.*
***8:** Select source of application version and click next*
***9:** Fill in environment information, 'environment name' and 'environment URL'. Click on Next. (Do not select any option from additional resources. Click next.)*
***10:** Review Configuration details and click next.*
***11:** Leave environment tags as it is and click next.*
***12:** Manage permissions and click next*
***13:** Review all the details and click launch.*
***14:** After uploading the '.war' file, wait till the health status reaches to ok.*
***15:** Review the monitoring status.*
***16:** Use the estimate distance and time to calculate the cab fares.*
***17:** Prediction is done on the basis of the results.*

The workflow of the proposed methodology is diagrammatically represented by figure 2.

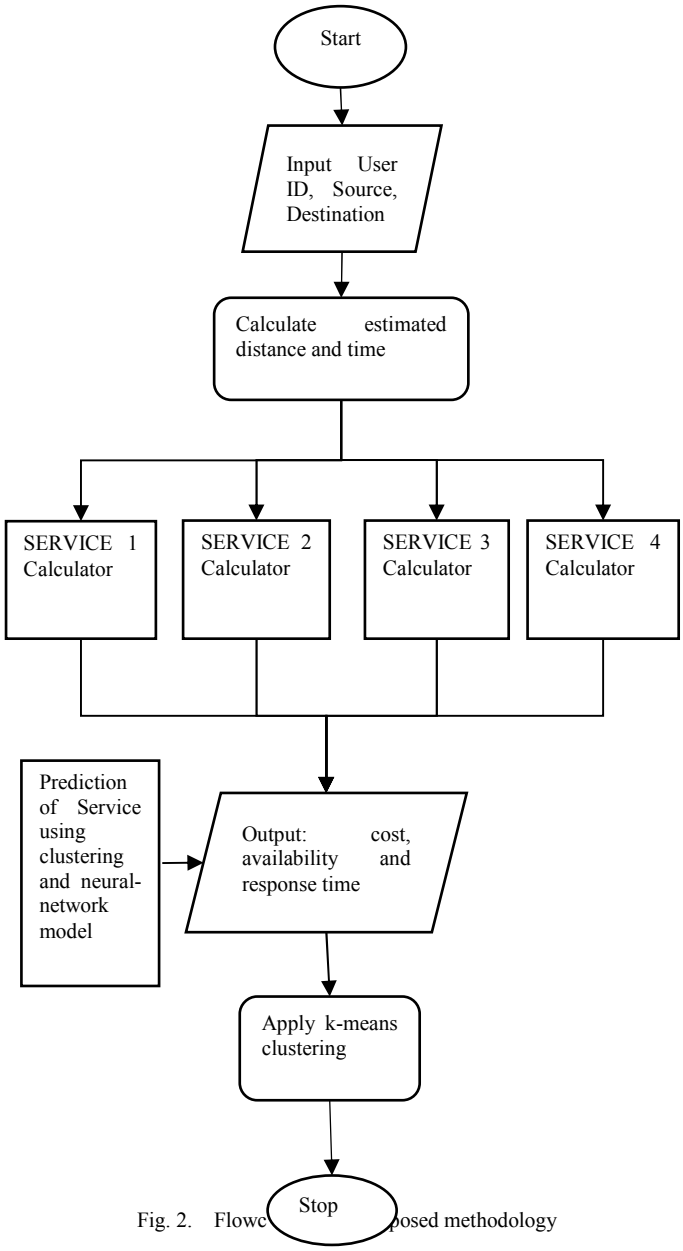

Fig. 2. Flowc posed methodology

The proposed model calculates the cost, availability and response time and then compares it with a service prediction model to propose the best cab service that should be taken by the customer. The architectural design can be seen in figure 3. The figure shows that the user inputs their starting point and destination point into the model. This data is then used in finding the estimated real-time distance and time using Google API. This is then send to different web services to find the approximate cost provided by each cab service.

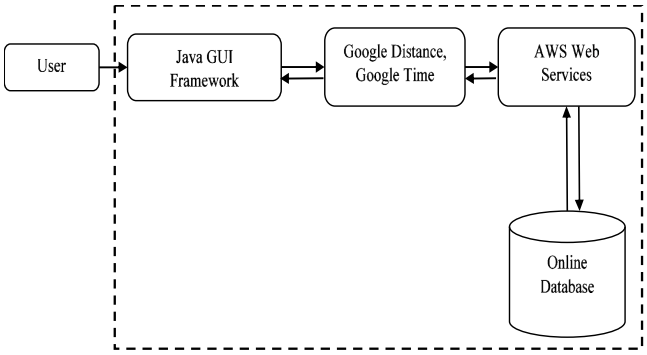

Fig. 3. Architecture Design

This cost is used as input in k-means clustering model shown by figure 4 which finds the clusters of cabs based on their services and minimum cost and hence finding the best cab service for the customer.

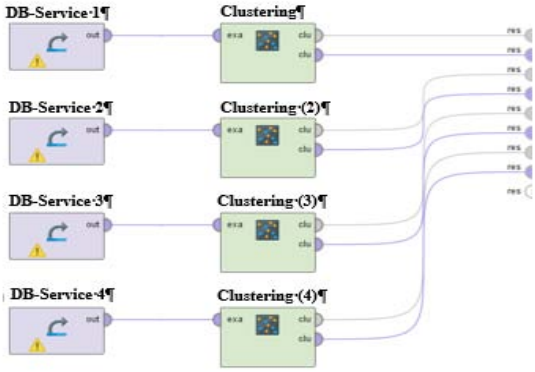

Fig. 3. K-means clusters in Rapid miner

IV. RESULTS

The results of the proposed methodology are shown by the following figures. The web service is successfully made and deployed on AWS which is easily accessible from anywhere.

Once the application is deployed the successful status is shown by the following figure.

The calculators for all four Service Providers are shown as given below.

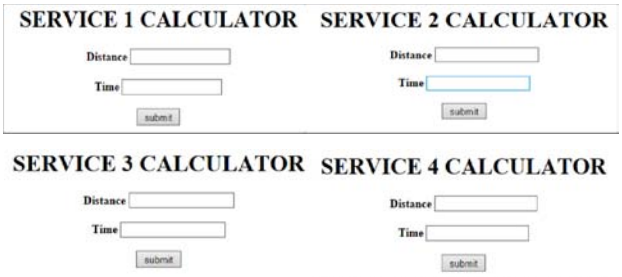

Fig. 5. Service Calculators for Input as Distance and Time

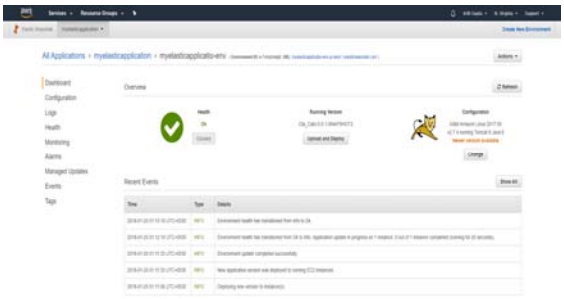

Fig. 6. Status of the application deployed

The analysis and monitoring can be carried out with the AWS analysis program. It shows response time, number of requests and various parametres that can be used for load balancing, increasing scalability and increasing elasticity of the model.

Fig. 7. Graphical representation of services in AWS

SERVICE 1:

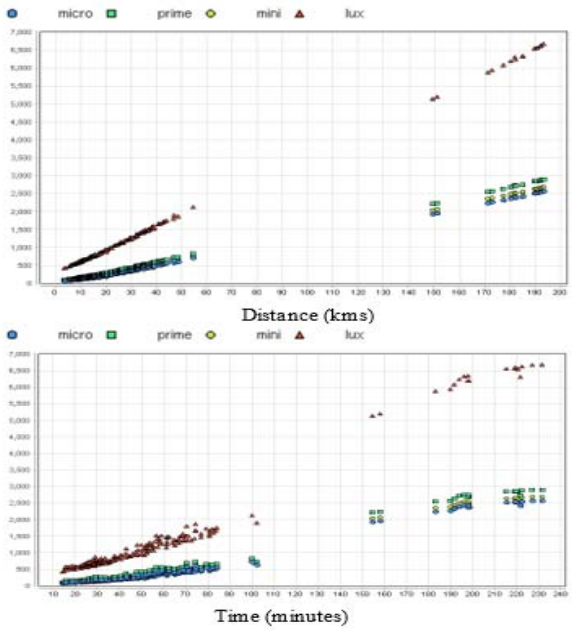

Fig. 8. Scatter Plots for Service 1 with Distance and Time

TABLE I. CLUSTER DISTRIBUTION FOR SERVICE 1

Nominal value	Absolute Count	Fraction
cluster_0	*114*	*0.5643564356435643*
cluster_1	*20*	*0.09900990099009901*
cluster_2	*68*	*0.33663366336633666*

The results given above show that three clusters are obtained namely cluster_0, cluster_1 and cluster_2 with absolute count as 114, 20 and 68 respectively in SERVICE 1.

SERVICE 2:

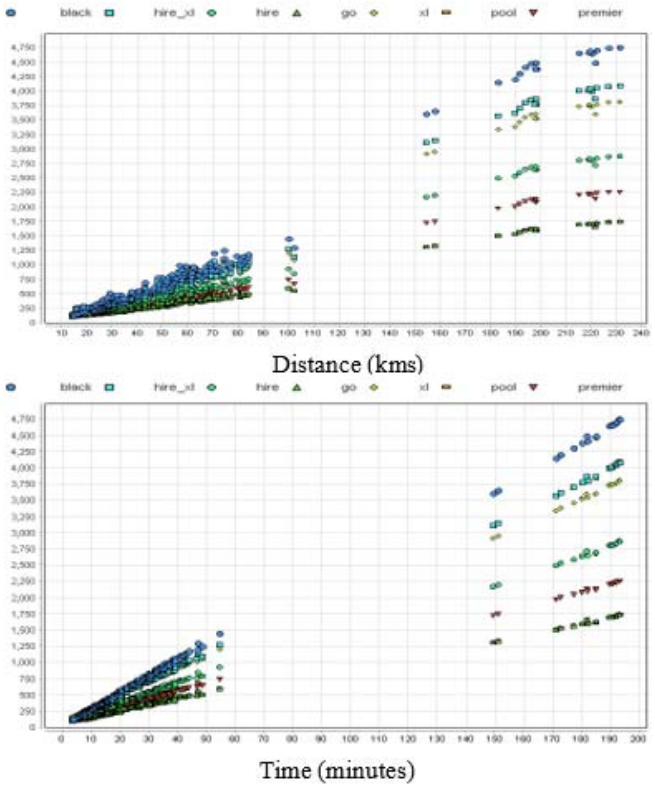

Fig. 9. Scatter Plots for Service 2 with Distance and Time

TABLE II CLUSTER DISTRIBUTION FOR SERVICE 2

Nominal value	Absolute Count	Fraction
cluster_0	*20*	*0.09900990099009901*
cluster_1	*110*	*0.5445544554455446*
cluster_2	*72*	*0.3564356435643564*

The results given above show that three clusters are obtained namely cluster_0, cluster_1 and cluster_2 with absolute count as 20, 110 and 72 respectively in SERVICE 2.

SERVICE 3

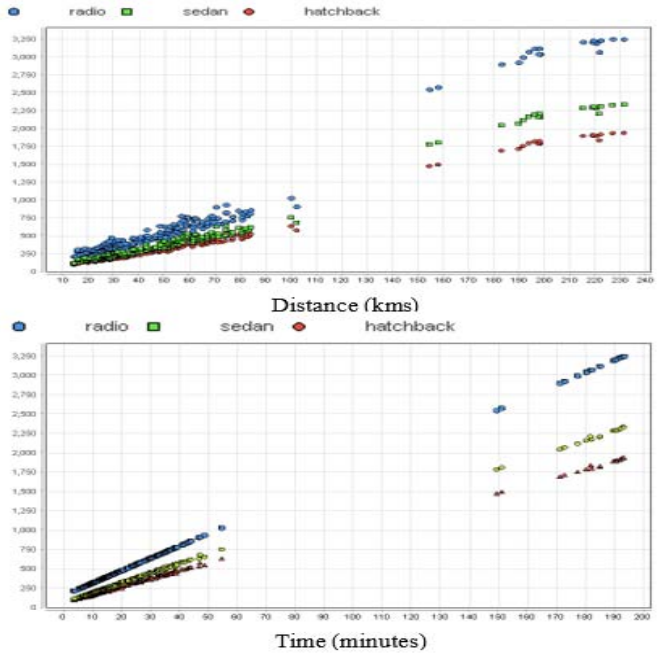

Fig. 10. Scatter Plots for Service 3 with Distance and Time

TABLE III CLUSTER DISTRIBUTION FOR SERVICE 3

Nominal_value	Absolute Count	Fraction
cluster_0	*110*	*0.5445544554455446*
cluster_1	*20*	*0.09900990099009901*
cluster_2	*72*	*0.3564356435643564*

The results given above show that three clusters are obtained namely cluster_0, cluster_1 and cluster_2 with absolute count as 110, 20 and 72 respectively in SERVICE 3.

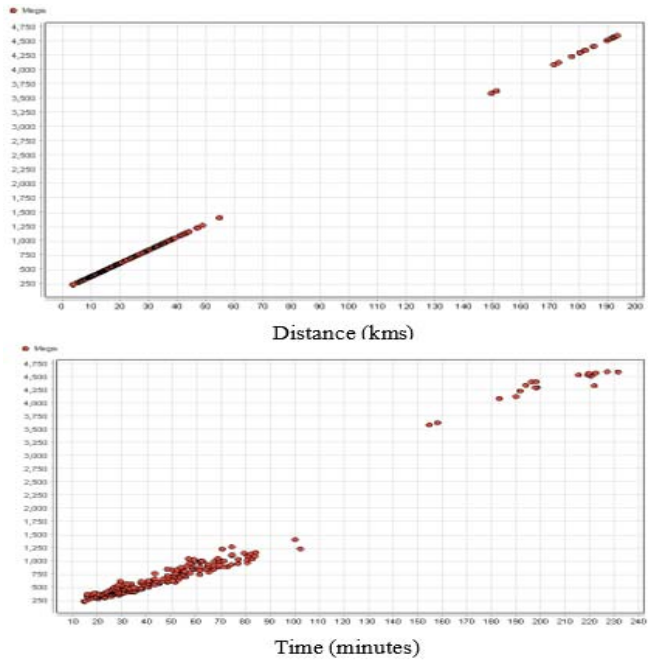

Fig. 11. Scatter Plots for Service 4 with Distance and Time

Table IV: cluster distribution for SERVICE 4

Nominal_value	Absolute Count	Fraction
cluster_0	*20*	*0.09900990099009901*
cluster_1	*71*	*0.35148514851485146*
cluster_2	*111*	*0.5495049504950495*

The results given above show that three clusters are obtained namely cluster_0, cluster_1 and cluster_2 with absolute count as 20, 71 and 111 respectively in SERVICE 4.

The multiple scatter plots are shown in figures 12 to 19 using k-means clustering algorithm. These results help in generating the availability, cost and response time of the services and with the help of a prediction model it can predict the best service to the customer.

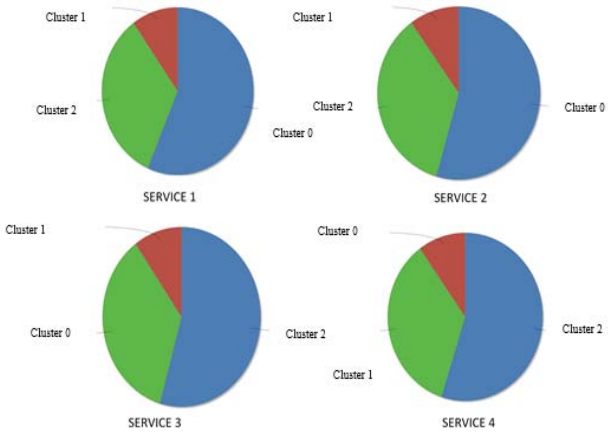

Fig. 12. Cluster distribution for each web service

The clusters formed after k-means clustering are cluster 0, cluster_1 and cluster_2. The clusters are formed in the order that shows cabs with services and their respective prices. cluster_0 shows the cabs with minimum price but no service and hence can be considered as the most frugal option. cluster_1 shows the cabs with average price and average service and hence can be considered as the optimum case. Service 2 is seen to have maximum cabs in this cluster and thus can be said to have the most optimized service. Cluster_1 shows cab with extremely luxurious services but with high prices and thus considered as most costly services. Thus, according to prices and services the clusters are formed and likewise the preferred cab can be assigned to the customer.

V. CONCLUSION AND FUTURE WORKS

IN concluding our work, we are stating that service composition and its internal structural design needs to be take care based on regular monitoring of service usage. In our proposed mathematical model aims at developing 4 web services in AWS using Amazon EC2 service and Elastic Beanstalk which will be further extended in composite service to reduce problem of finding best service from multiple service providers. Calculators are designed in order to calculate the approximate cost using distance and time as input using Google API. The clusters are formed using k-means

clustering in Rapidminer. On user request, we are analyzing them and find one of the best service using K-mean clustering. In our setup we demonstrate one example and find a service comes to be the most optimized one. The methodology in the project is highly competent and precise. It provides faster and better results as compared to existing methodologies. It can provide best service with inputs from any predictive model. In future, work can be done to improve the model with the help of increase in the number of policies and the load applied on the system

REFERENCES

[1] Kumbhare A. G., Simmhan Y., Frincu M., Prasanna V. K., *"Reactive Resource Provisioning Heuristics for Dynamic Dataflows on Cloud Infrastructure"*, IEEE Transactions on Cloud Computing, vol. 3, no.2, IEEE, 2015, Page No. 105-118.

[2] Hao H., Jiang H., Silva D. D., *"Enhancing Datacenter Resource Management through Temporal Logic Constraints"*, International Parallel and Distributed Processing Symposium, IEEE, 2017, Page No. 134-142.

[3] Yassin M., Ould-Slimane H. , Talhi C. and Boucheneb H., *"SQLIIDaaS A SQL injection intrusion detection framework as a service for SaaS providers"*, 4th International Conference on Cyber Security and Cloud Computing, IEEE, 2017, Page No. 163-170.

[4] Lee K. and Son M., *"DeepSpotCloud Leveraging Cross-Region GPU Spot Instances for Deep Learning"*, 10th International Conference on Cloud Computing, IEEE, 2017, Page No. 98-105.

[5] Kumbhare A. G., Simmhan Y., Prasanna V. K., *"PLAStiCC Predictive Look-Ahead Scheduling for Continuous Dataflows on Clouds"*, 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2014, Page No. 344-353.

[6] Zinno I., Casu F., Luca C. D., Elefante S., Lanari R., Manunta M., *"A Cloud Computing Solution for the Efficient Implementation of the P-SBAS DInSAR Approach",* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 3, 2017, Page No. 802-816.

[7] Khodadadi F., Calheiros R. N , Buyya R. K., *"A Data-Centric Framework for Development and Deployment of Internet of Things Applications in Clouds"*, IEEE 10th ISSNIP, Singapore, 2015, Page No. 1-6.

[8] Grimaldi D., Persico V., Pescape A., Salvi A., Santini S., *"A Feedback Control Approach for Resource Management in Public Clouds"*, IEEE GLOBECOM, 2015, Page No. 1-7.

[9] Arabnejad V., Bubendorfer K., Bryan N., *"Deadline Distribution Strategies for Scientific Workflow Scheduling in Commercial Clouds"*, IEEE/ACM 9th International Conference on Utility and Cloud Computing, 2016, Page No. 70-78.

[10] Bhathiya W., *"CloudAnalyst A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments"*, Distributed Computing Project, University of Melbourne, 2009, Page No. 433-659.