March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

Cloud Resource Allocation: A Novel Approach with Naïve Bayes Classifier & KDE

Abdulhamza A. Abdulkarim
Al-Mamoon University College, Department of Computer Engineering Techniques
Dr.hamza98@yahoo.co.uk

Article Info
Volume 83
Page Number: 3817 - 3832
Publication Issue:
March - April 2020

Article History
Article Received: *24 July 2019*
Revised: *12 September 2019*
Accepted: *15 February 2020*
Publication: *23 March 2020*

Abstract:
Cloud computing offers huge data processing with comfortable economy. In fact, it uses pay-per-use model that use like outsourcing of processing and storage equipment. Broadband and other network technologies make this idea into reality. On the other side, *Cloud* venders always try to use their resources in most efficient way that satisfy different customer and their heterogeneous requirements. As resources are always limited, Cloud venders multiplexed their resource among workloads. This switching can be performed by three main strategies including *Artificial Intelligence*, *Predictive Resource Allocation* and *Dynamic Resource Allocation* [1]. It is very clear that if cloud resource management system is enabled to predict the workload properly then Cloud system manage more efficiently. In following study, we develop and investigate, a model by using Naïve Bayes with Kernel Density Estimation. The evaluation of the model was impressive up 99.1% correct predictions.

Keywords: *Cloud Resource, Novel, Naïve Bayes, KDE, Classifier*

I INTRODUCTION

Cloud computing got popularity due to its low cost and maximum throughput over a decade. Cloud also expand its utilization in conventional utilizations models (IaaS, PaaS & SaaS) and non-conventional model XaaS [2] (so-called everything as service) with multiple deployment models. A typical Cloud environment can be setup as large data center where a collection of hardware and software resources are virtually always available up to unlimited extent . Cloud data center have SLA to its customers where they promise to not only provide QoS&QoE and also elasticity of the availability of resources [3] means as customer required more resources then Cloud must provide these resources without disturbing the equilibria of eco system and similarly at period end, the customer will pay the bill of resources used.

One of the major issues in Cloud, is to managing the resources in such proper way that every customer satisfied its needs and enjoy the pool of resources which are unlimited virtually. To show such character, Cloud system have to perform many operations including, switch resources, migrate services, balancing the load and disseminate the processing in Cloud nodes. Another major aspect is the internal management where cloud might have bear extra cost in terms of power consumptions and bandwidth consumption. Processing resources produce lot of heat and Cloud venders have to apply extra resources to bring system cool down [4].

Cloud resource requirements fell down in various workloads according to its weights. If we are able to predict about the weight of workload, then Cloud management system can perform its operation more conveniently. System can vacant the resources, identify the free nodes and shift the current workload on vacant nodes. Fortunately, there are multiple prediction techniques that can develop the models on the base previous log data and identify a sequence of processing as a class of weight [5]. If system is able to perceive future workload, it can be easily multiplex its resources and avoid over and under provisioning of resources.

Naïve Bayes is one the famous classifiers that easily build model over the given pattern and show to probability of a class. The Naive Bayes is probabilistic method that try to mark the data pattern with pre-defined classes. On the other hand, Kernel Density Estimation Function (KDEF) is non parametric probability density method for smoothing the data. KDEF use to improvise the NB and results were outstanding which will discuss later.

1.1. The Resource Allocation problem in Cloud Computing

As discuss earlier that Resource Allocation in Cloud is important of cloud computing in its own account. There are many parameters which elaborate the efficiency of Cloud management system. These dimensions are discussed in [4] [6] [1] [7]. In [6] elaborate the resource allocation problem in eight different methods including optimization objectives, design approach, target resource type, optimization methods, utility functions, processing mode, target instance and experimental setup. Another survey was done by [1] and they describe the resource allocation planning into two major categories, strategic and parametric. A comprehensive study was conducted by [7]. The focus was to explain the problem with multiple aspects including cost based, time based, bargaining based, profit based, SLA &QoS based, energy based, optimization based, nature inspired & bio inspired based, VM based, hybrid based and dynamic based. A specific content of machine learning based energy management in Cloud resources is discussed in [8]. All these surveys and studies are discussing following common and important points in Cloud resource management.

1.2. A general overview of Resource Allocation

Cloud work like a simple machine which take input, process it and return results. For outer world Cloud datacenter perform likes a single computing machine, it got memory, processing equipment, storage and network devices. Just like a simple computer, its operating system allows multiple process to execute. For this purpose, it performs scheduling, dispatching, mange program counter blocks and hold intermediate results with multiple schedule techniques like shortest job first (SJF), round robin etc. The main difference between a Cloud and single machine, is that Cloud have to server diversified types of workload with extensible pool of resources. For such kind of sophisticated services, Cloud have to setup a well-orchestrated monitoring and resource provisioning system. The monitoring system record granular events and continuous generate the usage log of resources. These logs are good source of anticipation and help to analyze the usage of Cloud. All the schemes and techniques that are define and propose to administrate the Cloud system, get aware of this information. A generalized method of such activities is presented in Fig1. All the events and activities are register to Cloud Resource Monitoring System. Resource Monitoring submit its data to Analyzer that find the over, under and well provisioned resources. In the meanwhile, another monitor, Workload monitoring system also submit its observations to analyzer as well. Analyzer perceive the requirements and available resource and find a well-managed, well-balanced and optimal assignments of resources to workload and invoke the Provision Manager.

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

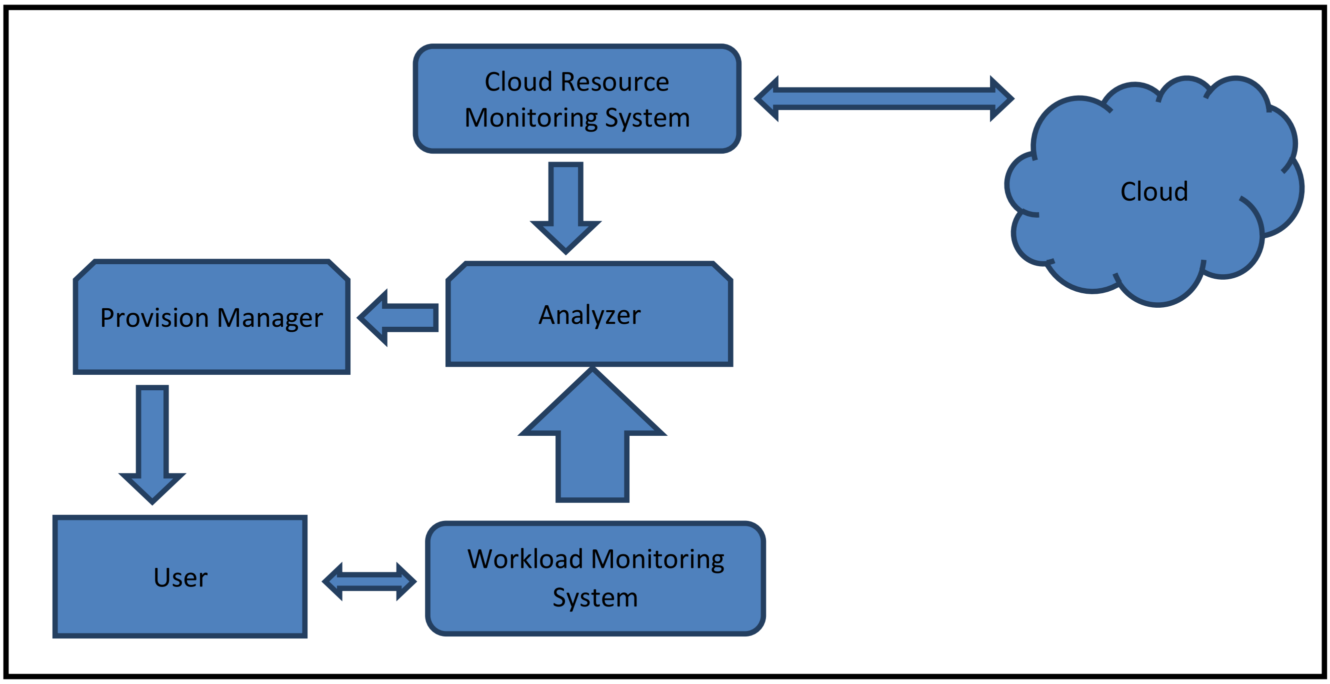

Figure 1. Cloud Resource Monitoring & Allocation System. A General View

All the efforts and methods for optimal resource utilization are applied to Analyzer. The proposed work is also an effort to update Analyzer make able it makes better predictions about the workload and responses of resources.

1.2.1. Energy efficiency

Energy is core resource of processing unit. All hardware run by using this energy including CPUs, GPUs, memory units, storage units and network devices. According to Gartner the energy requirements for datacenters are going to double for each five years [9]. This is not only the requirement of hardware utilization but also required reduce thermal emission in Cloud data centers. A proper cooling system is thermal aware and required to execute a specific level of heat. If over provision occurred in some data center nodes then might be cause extra production of heat and over utilization of cooling mechanism.

1.2.2. Load Balancing

Cloud resource provision system must be very sensitive about load of a node inside cloud and as node is under or over-utilization, the load balancer must invoke and perform its task [10]. Normally Cloud outlets execute over VMs and these VMs can be migrate to different nodes during execution. Multiple researches were conducted for load management such as Ray's algorithm work with [11]. Cost analysis for resource allocation was study by Gopal &Manvi [12]. Xu and Yu introduced a novel approach using game theory with multiple resource allocation. A nature inspired technique proposed by Young & et al [13]. This method uses Ant Colony Optimization to adjust efficient resource allocation.

1.2.3. QoS

QoS is directly related to user's experience of utilization of Cloud services. Quality of service can be disturbed by latency of access the resource [14], unavailability of resource [15], limitation of resource on demand [16] and security concerns [17]. The effect of bad QoS directly reduces the profit margins produce hardships for Cloud market share [18].

1.2.4. Availability

Cloud is elastic in demand. If a customer required more resources during execution of its cloudlet, then Cloud must provide them (according to basic definition of Cloud Computing by NIST [19]). Integer Programming based solution was proposed by [20]. Nathani proposed a technique that minimizes the denial of request by swapping deadline-critical jobs with less priority jobs [5]. Availability is more critical in mobile devices and a method of ensuring the resource availability in moving devices by Park et al [21]that implement the Markov chaining.

1.2.5. Workload Characterization

Any kind of workload can be assigned to cloud according to requirement. This workload can different in nature and weights. A study was produced by Acken, Sehgal&Sohoni [22] that describe different type of workload and also characterize it by their weights. A similar work was presented by Orzechowisky et al [23]. An overview of workload characterization and monitoring was published by Calzarossa et al [24].

1.3. The Naïve Bayes Model

The NB is probabilistic model that predict to identify the class of an instance on the base of simple probability [25]. NB model calculate the likelihood of unidentified label by given labeled instances. Let the training set contains the Labels $L = \{l_1, l_2, l_3, \ldots\}$ and $F = \{f_1, f_2, f_3, \ldots\}$ is set of features or feature vector.

The fundamental formulae for calculating probability are following according to labels and features

$$P\ (L \mid F) = \frac{P\ (F, L)}{P\ (F)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots. (1$$

$$P\ (L \mid F) = \frac{P\ (L)\ .\ P(F \mid L)}{P\ (F)} \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots. (2$$

Where P(L) is the possible label of given instance. In the similar way we can find the possible labels of all unknown instances. While P(F) is probability of feature on given label. Collective the probability can be ascertained as product of all features vector over label as shown in following equation.

$$P\ (L \mid f_1, f_2, f_3, \ldots f_n) = P\ (L)\ . \prod_{i=0}^{n} \quad P\ (f_i \mid L) \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots. (3$$

1.4. Kernel Density Estimation

KDE was initially proposed by Parzen [26] and later by Rosenblatt [27] independently for estimating probability density for random variables. The function use for smoothing problem in statistics. We can find the shape of function by using equation 3 on sample taken from distribution with unknown density.

$$f_h(x) = \frac{1}{nh} \sum_{i=1}^{n} \quad K_h\ (x - x_i) \quad \ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots. (4$$

$$n: Size\ of\ Sample$$
$$K: Kernal, a\ positive\ function$$
$$h: Smoothing\ paramter, always\ h > 0$$
$$K_h: Scaled\ Kernal$$
$$x_i: Instance\ of\ Sample$$

K_h can be defined from following expression

$$K_h(x) = \frac{1}{h}\ K(\frac{x}{h}) \quad \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots\ldots\ldots\ldots\ldots \ldots\ldots. (5$$

We can choose h as we want but the value of h can be increase bias

KDE is very common statistical operation that implements in many problems related to probability. Our research is also using probability so we implement the KDE as well using sklearn [28].

II LITERATURE REVIEW

Predictive workload characterization was discussed in all major surveys of Cloud resource allocation strategies. A review of recent proposed methods and techniques are reviewed in following section.

1. A novel approach was introduced by Wu et al [29]. They improvised the foundation

TEST Engineering & Management

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

NB algorithm that use for service resources classification. Additionally, it also implemented parallel programming model with hybrid of Hadoop and MapReduce.

2. Multiple Time Series approach was based in Hidden Markov Model (HMM) discovered by Khan, Yan, Tao and Anerousis [30]. This method was developed for VM workload prediction with various workload patterns.
3. The HMM was also used by Balaji, Kumar & Rao for enterprise workload. The approach was integrated by ARIMA (AutoRegressive Integrated Moving Average) [31].
4. To improve the prediction accuracy, Hu & et al develop a framework consist on three phases. Initially it uses a time series approach to monitoring data. Then Kalman filter was applied on data and finally a novel pattern matching algorithm was used for prediction. This method improves prediction but reduced automatic scaling delay [32].
5. By applying Markov modeling & Bayesian modeling on Google cluster data to find the better prediction was performed by John Panneerselvem and et al [33].
6. For robust prediction, fuzzy logic approach was introduced in [34]. The main focus was QoS, resource management and scalability.
7. A novel approach with time series was presented by Liu & et al [35]. This method uses different classification methods to assign labels to unidentified data patterns with mixed 0-1 Integer Programming. To prove the effectives of work, it was compared with ARIMA, SVM and Linear Regression. Google Cluster traces was used as dataset.
8. Predictive backpressure algorithm was introduced by Du and et al that implements backpropagation network to predict of video traffic [36]. The algorithm was handy to reduce the delay and increase accuracy.
9. Genetic Algorithm (GA) is another bio inspired algorithm that was presented by Tseng [37]. This algorithm was improvised with multiobjective to increase the utilization of CPU and memory in virtual and physical machines. This algorithm also showed some better results in energy consumption.
10. A hybrid solution was setup by Hadeel T. El Kassabi [38] that use collaborates monitoring and prediction-based adaptation. The framework increased the prediction accuracy and reduce the violations.
11. Power management issue was also tried to solve in [39]. It is a hierarchical framework that was built on Deep Reinforcement Learning. They address complicated control problem in large state space. An autoencoder was also a part of system that help to manage the high dimensional state space.
12. Martin Duggan and et al presented a new method of predicting of CPU consumption for a single-time step and multi-time step with help of Recurrent Neural Network [40]. The experiments reveal some good performance in prediction accuracy.
13. Subtractive-Fuzzy Clustering based Fuzzy Neural Network framework was published by Chen et al [35]. This work adopted some base predictor to organize some the ensembled model. The model predicts the demands of resources by cloud outlets.

These are some studies, researches and experiments were conduct in recent past for addressing problems in Cloud resource allocation

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

systems. The main focus of these studies was remained on classification and predicting the workload in Cloud. Our propose approach also novel and open some aspects and techniques of said issue.

2. *Proposed Scheme*

The main concentration of study is to develop a model that can predict more accurately the utilization of processing unit and memory on a labeled dataset that generated from real world scenarios. We choose Naïve Bayes classifier to obtain this objective. A Cloud node have to perform mixed amount execution over the data and these amounts could be change any time. A cloud management system cannot identify the pattern of upcoming events and execution requirements of workload but old patterns can help build a model for planning where Cloud administrations prepare the resources according to future loads.

Cloud monitoring system continuously generates the logs of workload that can be easily labeled with very low to very high-performance requirements. Our system will learn from this labeled data able to find the future patten of usage of resources. The overall system architecture is described in Fig 2. Before further elaboration, we will discuss following modules of our framework.

2.1. *Evaluation Scheme*

After establishing fundamental theorem, we can construct the hypothesis of evaluation. We will identify the three key performance metrics (m_i) consist on CPU consumption, Memory utilization and Response time. The parameters can be evaluated by above given model which is explained by equation 1, 2 & 3.

Table 2 Notations for measures

Measure	**Notation**
CPU Consumption Evaluation	m_1
Memory Utilization Evaluation	m_2
Response delay Evaluation	m_3

2.1.1. *CPU consumption evaluation (m_1)*

This is our first feature so now the equation will be formed as follows

$$P\ (m_1|f_1, f_2, \dots, f_n) = \frac{P\ (m_1)\ .\ P(f_1, f_2, \dots, f_n|}{P(f_1, f_2, \dots, f_n)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (6$$

2.1.2. *Memory utilization evaluation (m_2)*

$$P\ (m_2|f_1, f_2, \dots, f_n) = \frac{P\ (m_2)\ .\ P(f_1, f_2, \dots, f_n|}{P(f_1, f_2, \dots, f_n)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (7$$

2.1.3. *Response delay evaluation (m_3)*

$$P\ (m_3|f_1, f_2, \dots, f_n) = \frac{P\ (m_3)\ .\ P(f_1, f_2, \dots, f}{P(f_1, f_2, \dots, f_n)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (8$$

A combined form of 4, 5 & 6 is

$$P\ (m_i|f_1, f_2, \dots, f_n) = \frac{P\ (m_i)\ .\ P(f_1, f_2, \dots, f}{P(f_1, f_2, \dots, f_n)} \quad \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (9$$

2.2. *Features Schemes*

There is large list of parameters provided by different researchers that should be included for study as features or parameters [41]. Some of these parameters also further expand for new studies and remaining are basic parameters which are part of standard research.

TEST Engineering & Management

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

2.2.1. Frequency of cloudlets in unit time (f_1)

Cloudlet or job arrival is not uniform in any system so we took mean jobs in unit of time. We can also split the time in multiple windows of intervals from a minute to an hour. Multiple dataset was prepared for this purpose with different intervals. Expression for a unit of time can be described by mean formula.

$$f_i = \frac{\sum_1^n f}{n} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (10$$

2.2.2. Available Memory (f_2)

Cloud data center may consist on multiple nodes and each node may have heterogeneous resources. All cloudlets manage by the available memory to find the unknown amount of memory for given outlet we use following probability equation

$$P\ (L\ |f_2) = \frac{P\ (L)\ .\ P(f_2|\ L)}{P\ (f_2)}$$

2.2.3. Storage (f_3)

Secondary storage is an integral part of Cloud services. It not only uses for data but also increase the performance system in the form of virtual memory.

$$P\ (L\ |f_3) = \frac{P\ (L)\ .\ P(f_3|\ L)}{P\ (f_3)}$$

2.2.4. CPU Cores (f_4)

CPU is core device for data processing but multi core technology made this equipment more powerful. It is also important that all cores of a processor not equally loaded with job but a greater number of cores made it possible candidate of job.

$$P\ (L\ |f_4) = \frac{P\ (L)\ .\ P(f_4|\ L)}{P\ (f_4)}$$

2.2.5. Core Clock Speed (f_5)

As CPU have high clock speed then it is able process more data while a core of CPU has similar clock speed.

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \ P\ (L\ |f_5) = \frac{P\ (L)\ .\ P(f_5|\ L)}{P\ (f_5)} \ (11$$

2.3. Workload Classes

Responses are converted to four labeled classes.

Table 3 Responses and Class Labels

Response (%)	Class Label
0-25	Very Low
26-50	Low
51-75	Medium
76-100	High

2.4. Experimental setup

Table 4 Experimental Setup Configuration

Programming Tools	ı, Numpy, Scikit-learn 0.22.1
Operating System	›ws 10 64bit
CPU	A6-5200, 4 Cores, 2.00 GHz
Memory	8GB
System	›tebook

Application was written in Python while KDE and NB are available in Scikit-learn. The architecture of application as following.

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

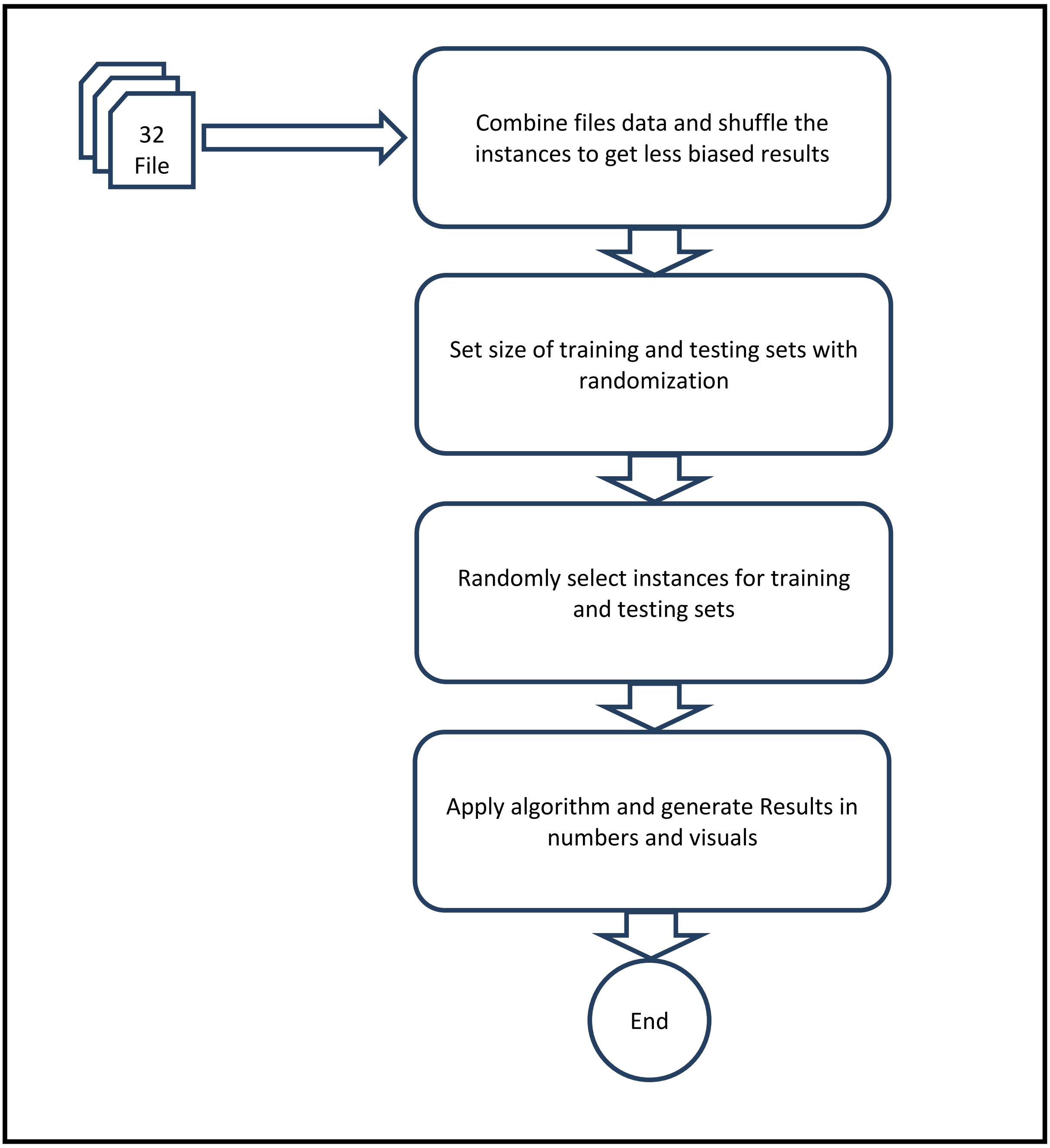

Figure 2. Application Architecture

2.5. *Dataset*

Dataset was collected from a private Cloud data center with heterogenous configuration. 20 physical machines where deployed at a single station. The overall configuration as follow

Table 5 Cloud Configuration

Total Physical Machines	20
Total Cores	108
Total Memory	256GB
Total Storage	32 TB

Total 32 files of data were obtained from 10 minutes span were recorded during dataset generation. 24 spells of the dataset produced during high demand hours and rest of the 8 were

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

recorded during low demand period. We also want to record any change in behavior of system during high and low demand. Load balancer was installed that equally distribute the workload over the all Cloud nodes. Each spell records the logs of features described in section 4.4 in rows in ".csv" file. Each file contains the instances created during hour. Sum of recorded instances is 136,344, while 124,659 belongs to peak hours while 11,685 rows were recorded low demand hours. It is considered that all nodes where equally loaded during processing.

2.6. Evaluation Parameters

It is most important task to test our proposed technique over the identified parameters of machine learning. These metrics might be in contradiction to each other but we have found a reasonable value during the test.

2.6.1. Accuracy

Prediction is the outcome of any machine learning algorithm. The correctness of prediction is considered to be the most important dimension because all of the next decisions are making over this prediction [42]. The accuracy can be evaluated in mathematical form.

$$A(\%) = \frac{Total\ Correct\ Predictions}{Total\ Predictions} \times 100 \qquad (15$$

2.6.2. Confusion Matrix

Confusion matrix describes the total correct and false outputs of a machine learning schemes. Confusion matrix build on 4 categories of outcome. True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). If outcomes lie in TP or FN then it means our system classify the instance correctly but it lies in TN or FP then outcome belongs to opposite class [43].

Positive (Number of positive instances assigned to positive class)	Negative (Number of negative instances to positive class)
False Positive (Number of positive instances assigned to negative class	False Negative (Number of negative instances to negative class)

III RESULTS

After generating records from machines, the log files migrated to experiment machine and application developed in Python executed and generate following results. The application split the data in training and testing sets and then apply NB with KDE over training dataset. Then these results matched to testing dataset. The size of training and testing data set randomly between 30% to 70% in three different folds. Application also calculate the accuracy, confusion matrix, comparison matrix and generate graphs for data visualizations.

As mentioned, total 136,344 instances were selected, the application set the split ratio 33% (44,994 instances) as testing set and 67% (91,350) as testing set. After running the fold, the system showed overall 96.16% accuracy. The confusion matrix shows 95.00% accuracy in "Very Low" label and remain 2.69% instances were selected as "Low" and 2.31% in medium. While no instance was selected as "High". There a fractional difference between CPU consumption, memory utilization and response time.

For next run, system selected 52% (71,638) as training and 48% (64,706) as volume of testing set. this time we got increment in the accuracy with about 96.49% overall. It also small increment in the incorrectly labeling the class "Very Low" with "Low" (2.69%) but incorrection in medium remain at 0.44%.

For the third run the system choose 68% as training and 32% as testing data set. We got total accuracy more high and better confusion matrix. 99.1% overall accuracy and correct marking of "Very Low" class moves to 98.22%. While incorrect labeling remain with "Low" at 1.78%

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

and no instances were labeled with "Medium" or "High". Similarly, all other classes got increment accuracy as well.

Dataset was taken in distinct forms of peak demand hours and low demand hours. The scheme was also tested with both parameters as well. There was slight change in response from system.

For more fair results, the app first collected all the data items from files and merge and shuffle them, so the difference between high and low demand hours remain same in behavior.

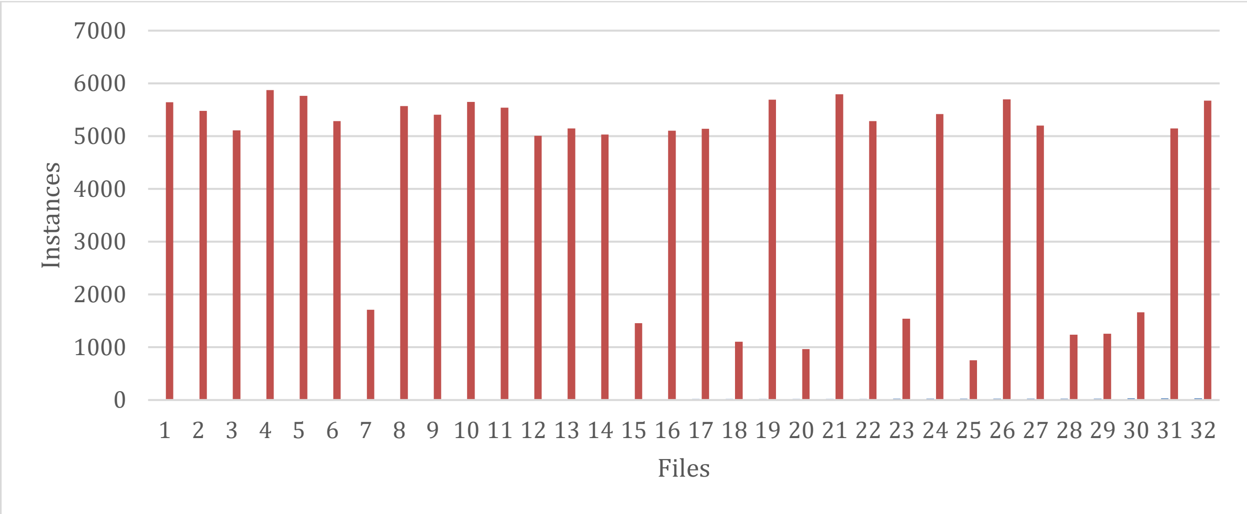

Figure 3. Files and Containing Instances

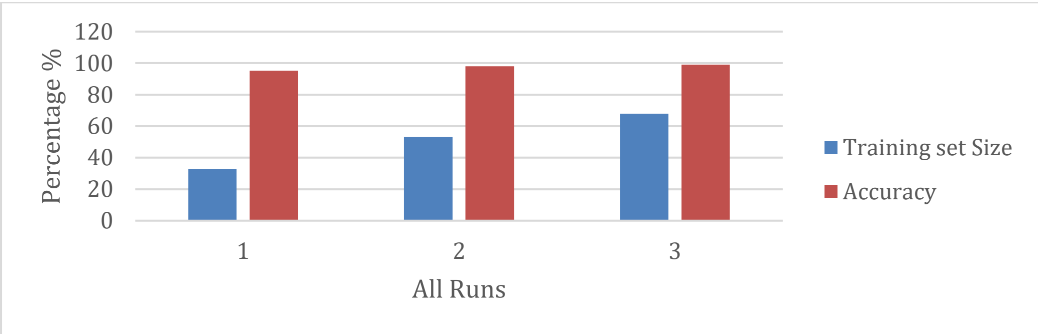

Figure 4. Comparison of All Three Runs with Accuracy

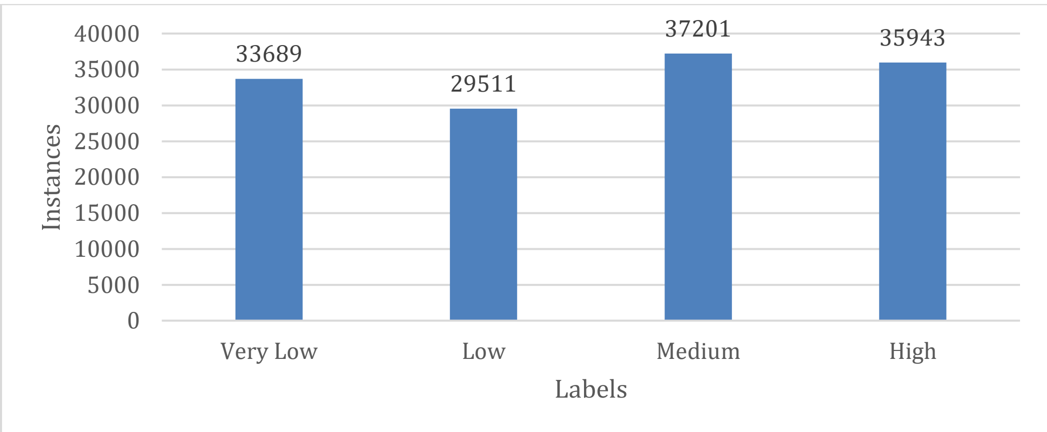

Figure 5. Labels and number of Instances in each Label

Following graph shows the effectiveness of first run. 33% instances were selected as training set. Further breakup with labels as follows

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

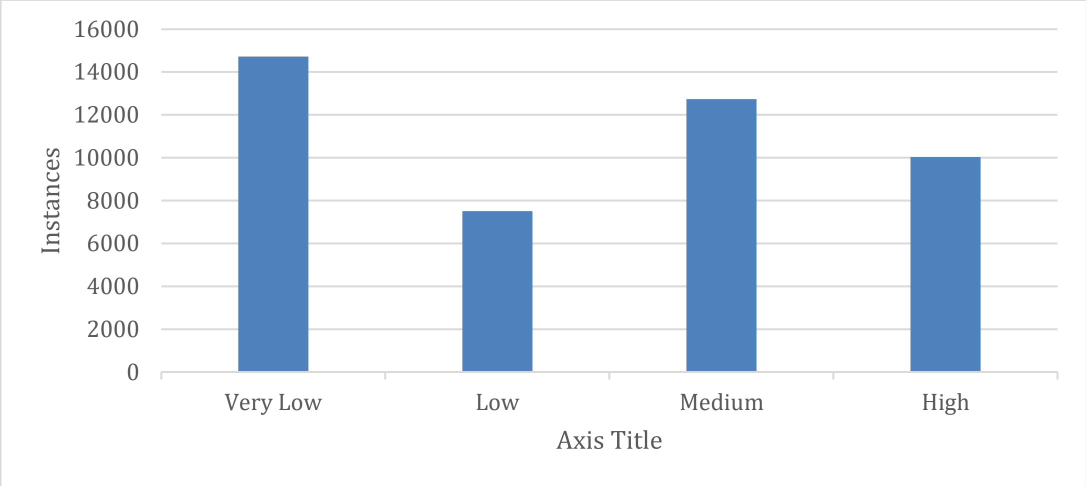

Figure 6. Instances in each Label of First Run

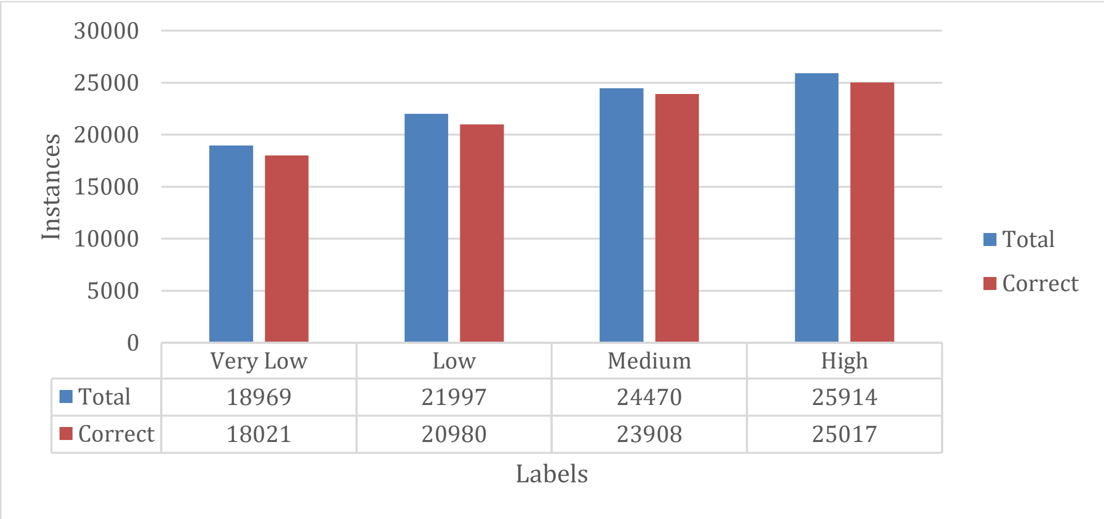

Figure 7. Comparison of Total and Correctly identified instances in first Run

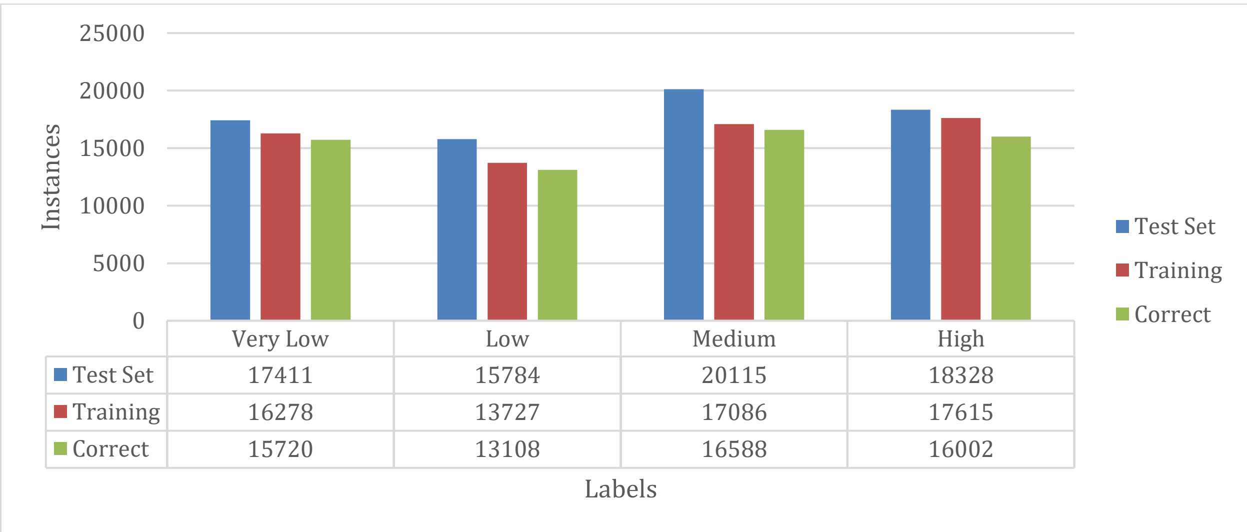

Figure 8. Comparison of Training, Testing and Correctly identified instances in 2nd Run

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

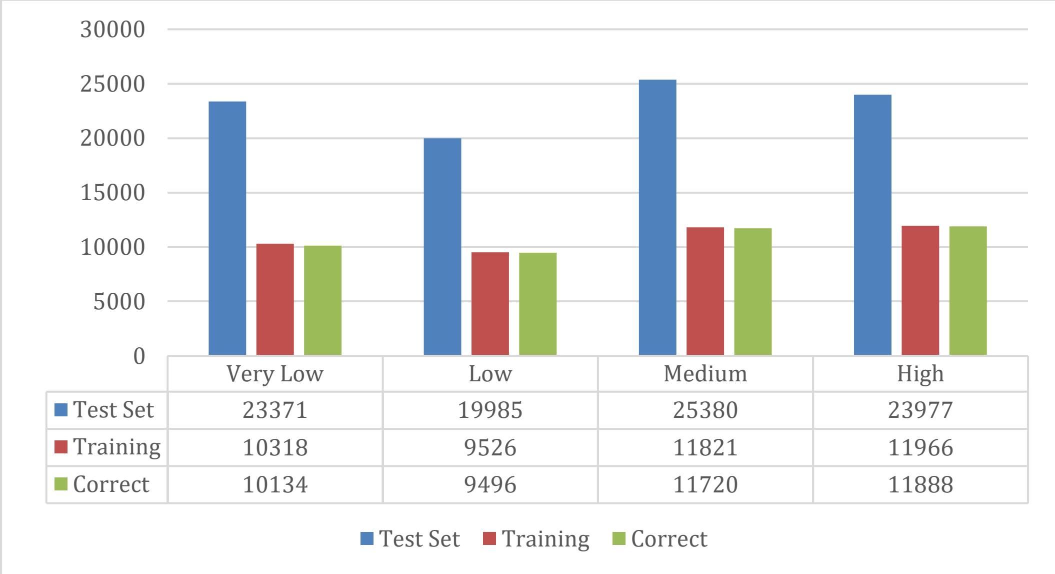

Figure 9. Comparison of Training, Testing and Correctly labeled instances in 3rd Run

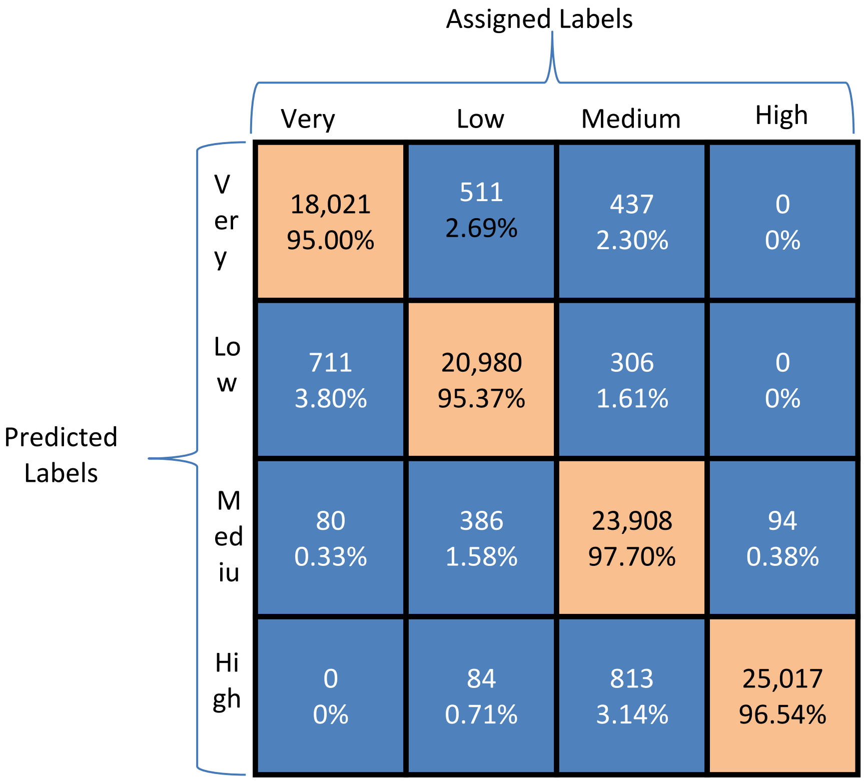

Figure 10. Overall predictions in 1st Run

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

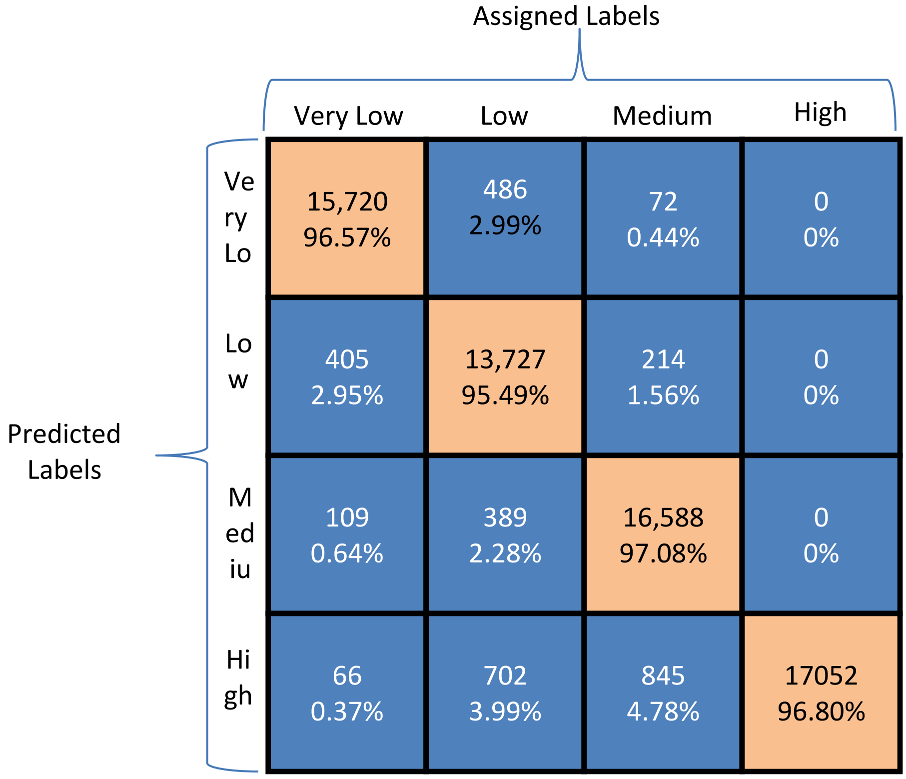

Figure 11. Overall predictions in 2nd Run

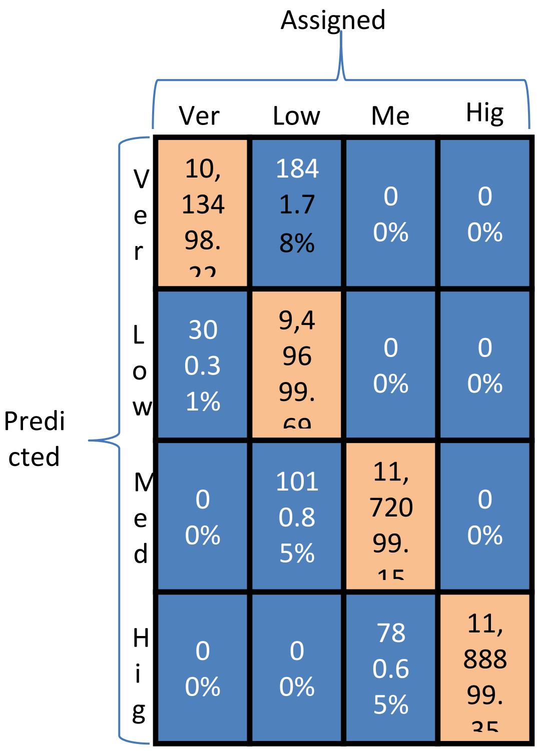

Figure 12. Overall predictions in 3rd Run

IV CONCLUSION

Dataset was collected from different frame of time and multiple nodes of a single private. Similarly, the dataset was also varied during the procedure by the application that diversify the selection parameters in instances and also in the volume of training and testing sets. The things are

March - April 2020
ISSN: 0193 - 4120 Page No. 3817 - 3832

very clear about set parameters. We obtained high accuracy in general and confusion matrix. the prediction model also verified over the heterogenous configuration of Cloud nodes. Overall results were satisfactory and show that NB model with KDE is good candidate along with other machine learning schemes that were used as prediction model in Cloud resource allocation.

V LIMITATION & FUTURE WORK

The dataset and experiments were extracted and conducted in very control environment so the robustness of results is one the primary limitations. Similarly, not all the parameters of Cloud system were under-consideration for detail analysis and design for large level applications. Another aspect is not addressed fully, the machine learning performance metrics.

VI REFERENCES

[1]. S. H. H. Madni, M. S. A. Latiff, Y. Coulibaly, and S. i. M. Abdulhamid, "Recent advancements in resource allocation techniques for cloud computing environment: a systematic review," *Cluster Computing*, vol. 20, no. 3, pp. 2489-2533, 2017/09/01 2017, doi: 10.1007/s10586-016-0684-4.

[2]. Y. Duan, G. Fu, N. Zhou, X. Sun, N. C. Narendra, and B. Hu, "Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends," in *2015 IEEE 8th International Conference on Cloud Computing*, 27 June-2 July 2015 2015, pp. 621-628, doi: 10.1109/CLOUD.2015.88.

[3]. Y. Al-Dhuraibi, F. Paraiso, N. Djarallah, and P. Merle, "Elasticity in Cloud Computing: State of the Art and Research Challenges," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 430-447, 2018, doi: 10.1109/TSC.2017.2711009.

[4]. A. Hameed *et al.*, "A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems," *Computing*, vol. 98, no. 7, pp. 751-774, 2016/07/01 2016, doi: 10.1007/s00607-014-0407-8.

[5]. A. Nathani, S. Chaudhary, and G. Somani, "Policy based resource allocation in IaaS cloud," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 94–103, 2012, doi: 10.1016/j.future.2011.05.016.

[6]. A. Yousafzai *et al.*, "Cloud resource allocation schemes: review, taxonomy, and opportunities," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 347-381, 2017/02/01 2017, doi: 10.1007/s10115-016-0951-y.

[7]. S. Singh and I. Chana, "A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges," *Journal of Grid Computing*, vol. 14, no. 2, pp. 217-264, 2016/06/01 2016, doi: 10.1007/s10723-015-9359-2.

[8]. M. Demirci, "A Survey of Machine Learning Applications for Energy-Efficient Resource Management in Cloud Computing Environments," in *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 9-11 Dec. 2015 2015, pp. 1185-1190, doi: 10.1109/ICMLA.2015.205.

[9]. Y. N. Alanazi Rayan, "Resource Prediction for Big Data Processing in a Cloud Data Center," *IEIE Transactions on Smart Processing & Computing*, no. 7, 7, p. 10, 2018.

[10]. K. A. Nuaimi, N. Mohamed, M. A. Nuaimi, and J. Al-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms," in *2012 Second Symposium on Network Cloud Computing and Applications*, 3-4 Dec. 2012 2012, pp. 137-142, doi: 10.1109/NCCA.2012.29.

[11]. S. Ray and A. d. Sarkar, "Resource Allocation Scheme in Cloud Infrastructure," in *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, 15-16 Nov. 2013 2013, pp. 30-35, doi: 10.1109/CUBE.2013.16.

[12]. G. K. Shyam and S. S. Manvi, "Resource allocation in cloud computing using agents," in *2015 IEEE International Advance Computing Conference (IACC)*, 12-13 June 2015 2015, pp. 458-463, doi: 10.1109/IADCC.2015.7154750.

[13]. Y. Moon, H. Yu, J.-M. Gil, and J. Lim, "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments," *Human-centric Computing and Information Sciences,* vol. 7, no. 1, p. 28, 2017/10/09 2017, doi: 10.1186/s13673-017-0109-2.
[14]. J. Z. Wei, Ao. Yuan, Jie. Yang, Fangchun, "AIMING: Resource Allocation with Latency Awareness for Federated-Cloud Applications," 2018.
[15]. Y. Hu, J. Wong, G. Iszlai, and M. Litoiu, "Resource provisioning for cloud computing," presented at the Proceedings of the 2009 Conference of the Center for Advanced Studies on Collaborative Research, Ontario, Canada, 2009. [Online]. Available: https://doi.org/10.1145/1723028.1723041.
[16]. B. G. Batista *et al.*, "Performance Evaluation of Resource Management in Cloud Computing Environments," *PLOS ONE*, vol. 10, no. 11, p. e0141914, 2015, doi: 10.1371/journal.pone.0141914.
[17]. A. Behl and K. Behl, "An analysis of cloud computing security issues," in *2012 World Congress on Information and Communication Technologies*, 30 Oct.-2 Nov. 2012 2012, pp. 109-114, doi: 10.1109/WICT.2012.6409059.
[18]. D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing: modeling techniques and their applications," *Journal of Internet Services and Applications,* vol. 5, no. 1, p. 11, 2014/09/25 2014, doi: 10.1186/s13174-014-0011-3.
[19]. P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.
[20]. G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing,* vol. 54, no. 2, pp. 252-269, 2010/11/01 2010, doi: 10.1007/s11227-009-0318-1.
[21]. J. Park, H. Yu, K. Chung, and E. Lee, "Markov Chain Based Monitoring Service for Fault Tolerance in Mobile Cloud Computing," in *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, 22-25 March 2011 2011, pp. 520-525, doi: 10.1109/WAINA.2011.10.
[22]. W. D. Mulia, N. Sehgal, S. Sohoni, J. M. Acken, C. L. Stanberry, and D. J. Fritz, "Cloud Workload Characterization," *IETE Technical Review*, vol. 30, no. 5, pp. 382-397, 2013/09/01 2013, doi: 10.4103/0256-4602.123121.
[23]. M. Orzechowski, B. Balis, K. Pawlik, M. Pawlik, and M. Malawski, "Transparent Deployment of Scientific Workflows across Clouds - Kubernetes Approach," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion),* 17-20 Dec. 2018 2018, pp. 9-10, doi: 10.1109/UCC-Companion.2018.00020.
[24]. M. C. C. L. M. D. Tessera, "Workload Characterization: A Survey Revisited," *ACM Comput. Surv.*,vol. 48, 2016.
[25]. M. E. Maron, "Automatic Indexing: An Experimental Inquiry," *J. ACM*, vol. 8, no. 3, pp. 404–417, 1961, doi: 10.1145/321075.321084.
[26]. E. Parzen, "On Estimation of a Probability Density Function and Mode," *Ann. Math. Statist*, vol. 33, no. 3, p. 11, 1962.
[27]. M. Rosenblatt, "Remarks on Some Nonparametric Estimates of a Density Function," *Ann. Math. Statist*, vol. 3, p. 5, 1956.
[28]. "Density Estimation. sklearn.neighbors.KernelDensity." [Online]. Available: https://scikit-learn.org/stable/modules/density.html.
[29]. Q. Wu, M. Cui, M. Zhang, R. Zheng, and Y. Lou, "A cloud service resource classification strategy based on feature similarity," *Journal of Networks*, vol. 9, no. 11, p. 2987, 2014.
[30]. A. Khan, X. Yan, S. Tao, and N. Anerousis, "Workload characterization and prediction in the cloud: A multiple time series approach," in *2012 IEEE Network Operations and Management Symposium*, 16-20 April 2012 2012, pp. 1287-1294, doi: 10.1109/NOMS.2012.6212065.
[31]. M. Balaji, C. Aswani Kumar, and G. S. V. R. K. Rao, "Predictive Cloud resource management framework for enterprise

workloads," *Journal of King Saud University - Computer and Information Sciences,* vol. 30, no. 3, pp. 404-415, 2018/07/01/ 2018, doi: https://doi.org/10.1016/j.jksuci.2016.10.005.

[32]. H. Yazhou, D. Bo, P. Fuyang, and W. Dongxia, "Workload prediction for cloud computing elasticity mechanism," in *2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 5-7 July 2016 2016, pp. 244-249, doi: 10.1109/ICCCBDA.2016.7529565.

[33]. J. Panneerselvam, L. Liu, N. Antonopoulos, and Y. Bo, "Workload Analysis for the Scope of User Demand Prediction Model Evaluations in Cloud Environments," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, 8-11 Dec. 2014 2014, pp. 883-889, doi: 10.1109/UCC.2014.144.

[34]. L. Zhang, Y. Zhang, P. Jamshidi, L. Xu, and C. Pahl, "Service workload patterns for Qos-driven cloud resource management," *Journal of Cloud Computing,* vol. 4, no. 1, p. 23, 2015/12/15 2015, doi: 10.1186/s13677-015-0048-2.

[35]. C. Liu, C. Liu, Y. Shang, S. Chen, B. Cheng, and J. Chen, "An adaptive prediction approach based on workload pattern discrimination in the cloud," *Journal of Network and Computer Applications,* vol. 80, pp. 35-44, 2017/02/15/ 2017, doi: https://doi.org/10.1016/j.jnca.2016.12.017.

[36]. J. Du, C. Jiang, Y. Qian, Z. Han, and Y. Ren, "Resource Allocation With Video Traffic Prediction in Cloud-Based Space Systems," *IEEE Transactions on Multimedia,* vol. 18, no. 5, pp. 820-830, 2016, doi: 10.1109/TMM.2016.2537781.

[37]. F. Tseng, X. Wang, L. Chou, H. Chao, and V. C. M. Leung, "Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm," *IEEE Systems Journal,* vol. 12, no. 2, pp. 1688-1699, 2018, doi: 10.1109/JSYST.2017.2722476.

[38]. H. El-Kassabi, M. A. Serhani, R. Dssouli, N. Al-Qirim, and I. Taleb, "Cloud Workflow Resource Shortage Prediction and Fulfillment Using Multiple Adaptation Strategies," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, 2-7 July 2018 2018, pp. 974-977, doi: 10.1109/CLOUD.2018.00149.

[39]. N. Liu *et al.*, "A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 5-8 June 2017 2017, pp. 372-382, doi: 10.1109/ICDCS.2017.123.

[40]. M. Duggan, K. Mason, J. Duggan, E. Howley, and E. Barrett, "Predicting host CPU utilization in cloud computing using recurrent neural networks," in *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*, 11-14 Dec. 2017 2017, pp. 67-72, doi: 10.23919/ICITST.2017.8356348.

[41]. P. Langley, "Selection of relevant features in machine learning," in *Proceedings of the AAAI Fall symposium on relevance*, 1994, vol. 184, pp. 245-271.

[42]. J. B. PhD, "How To Know if Your Machine Learning Model Has Good Performance," 2018. [Online]. Available:

[43]. S. Narkhede, "Understanding Confusion Matrix," 2018. [Online]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62.

[44]. Weston, S.M., Martin, E.D., Shippen, M.E., Kraska, M.F. and Curtis, R.S., 2017. Parents with Serious Mental Illness Served by Peer Support Specialists. International Journal of Psychosocial Rehabilitation, 21(2).

[45]. Griffiths, C.A., 2017. Determinates of mental illness severity. International Journal of Psychosocial Rehabilitation, 21(2).

[46]. Lau, H.Y., 2017. Family Therapy and Cognitive Behavioral Therapy for a Case with Co-morbidity of Depression and General Anxiety Disorder in Hong Kong–A Single Case Study. International Journal of Psychosocial Rehabilitation. Vol 21 (2) 21, 30.