Auto Selection of Clustering Techniques Using Cluster Validations for Cloud Log Analysis

Sreekanth D, Gladston Raj S.

ABSTRACT--- The work identified the challenges and requirements to select the right algorithm for clustering to detect the anomalies in the cloud Log Data. Having identified the gaps in the existing research in the area of cloud log analysis, it looks forward to providing a meaningful model for handling the problem dealing with different datasets like logs of OpenStack cloud, Hadoop and Spark. The system is capable to choose the right algorithm for the dataset comes up for the analysis. Here it is used the cluster validation techniques to select the right algorithm and based on the threshold values it is possible to differentiate the data in the dataset.

Keywords—Audio and Image, GPS, GPRS, GSM, Sensors.

I. INTRODUCTION

One of the effective ways of partitioning the datasets into a set of groups can be achieved through the unsupervised machine learning technique, clustering[1]. It can effectively make the usage of grouping whenever the data comes up without any labels[2]. In most of the cases, system can't expect that the unstructured data generating all over the world can be easily categorized. In that case, it is impossible to apply the supervised machine learning techniques for making the categorization.[3]

Unsupervised machine learning methods may deploy to fix this issue by clubbing the data into clusters[4]. It is helpful when it is required to create a snapshot of the data, and it can be labeled based on the formation of groups. It is treated as the initial stage towards implementation of supervised learning techniques for making predictions.[5]

A typical application may used for unsupervised learning is to deal with the massive amount of data that which generates all over the world.[6] The technique can be productively used for monitoring the system logs of various environments and can implement security measures towards protection of assets of every organization.[7]

II. CLUSTER VALIDATION

1) In general, the clustering validations can be branded into four major modules. [8]
2) Relative clustering validation:Works by evaluating the clustering structure by applying various parameters values in the same algorithm. Mostly it uses to find out the optimal quantity of clusters like in Elbow Method for K-Means and Dendrogram for Hierarchal Clustering etc.
3) External clustering validation: It is method can be used for comparing the cluster results with already existing dataset labels. The performance can be checked be deploying the confusion metrics by evaluating True Positives (TP), True Negatives (TN), False positive (FP) and False Negatives(FN).
4) Internal clustering validation:This method is used to evaluate the fineness of the clustering process using internal information. It is used to find out the optimal number of clusters and can identify the best fit clustering algorithm to the dataset provided.
5) Clustering stability validation:It is a special version can be used for the internal validation, and it evaluates the reliability of the algorithm by comparing it with clusters obtained after each column is removed.

III. METHODOLOGY

The model which use here is the Internal Clustering Validation since it is going to deal with data without having any of the prior information. The Fig-1 depicts the cluster validation model

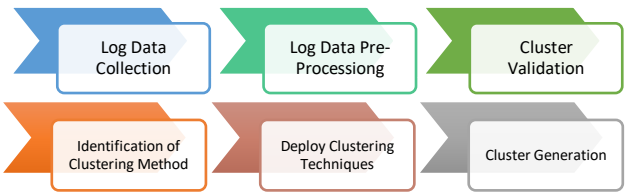

Fig-1: Cluster Validation Model

A. Internal Clustering Validation

The Internal Clustering Validation includes Connectivity, Silhouette Width and Dunn Index. The connectivity specifies the degree of relatedness of clusters determined by K-Nearest Neighbors.[5] The range of connectivity is from zero to infinity, and it is always better to be minimized. The methods silhouette width and Dunn Index shows the compactness and separation of clusters. The silhouette width is the average every observations value. The range of the value lies in between 1 and -1. Dunn index is the ratio

Revised Manuscript Received on June 10, 2019.

Sreekanth D,Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India. (sreekanth.dhanapal@gmail.com)
Dr.Gladston Raj S., Assistant Professor and Head, Department of Computer Science, Govt. College, Nedumangad, Trivandrum, Tamil Nadu, India.. (gladstan@rediffmail.com)

between the smallest distances between observations to the Intra-cluster distance. The range of lies in between 0 to Infinity. Table 1 gives a summary of internal cluster validation.

Table .1: Internal Clustering Validation

Validation	*About*	*Range*	
Connectivity	The degree of Relevancy of the Clusters using K-NN	0	Infinity
Dunn Index	Smallest Distance to the Intra-Cluster	0	Infinity
Silhouette Width	The degree of Confidence in a Cluster	-1	1

B. Kernel PCA

High dimensions take much time to process, and it will be badly affecting the performance of the proposed model. Mostly the log analysis is required to prevent the security breaches, and high-performance models are essential in such a case. System Logs from various sources are coming up with heterogeneous and with multi-well relevant dimensions.[7] So, applying dimensionality reduction techniques are very much essential and, in the case, if the values come are not in the linearly separable form may lead to the problem of obtaining the excellent group of clusters. In the research, have used Kernel Principal Component Analysis for the dimensionality reduction. The figures obtained after the process on various datasets depicts that it is tough to deploy the linearly separable model with the required dimensions[9].

C. Cluster Selection and Performance Evaluation

It is used the Internal Measures for Cluster Validation because the model proposes data processing as separate chunks. The Internal Measures for Clustering Validation works by dividing the dataset as clusters of objects such as objects in the same group are like the maximum possibility and objects in the different groups will be highly dissimilar[8][10].

The datasets used for the model building are taken from the Loghub at Github. Loghub preserves a group of system logs, which are freely available for investigation purposes. Some of the records are production information released from preceding researches, while some others are collected from real systems in their lab environment. They assure no modifications made to the Log Data. All these logs amount to nearly 100 GB in total. It is used only the available 2000 lines of records from various datasets for the investigation[11].

D. Identification of Clusters with Anomalies

Here the system calculates the shortest, furthest points of the boundaries where the cluster elements are present. Then will compare it with the minimum and maximum threshold of normal category of clusters[3]. In the case if any of the points come out of the threshold limit, that cluster will be treated as an anomaly[12]. Also provides the facility to set the new threshold levels based on the reinforcement learning of the system. In this research the threshold level has fixed by analyzing the normal behavior OpenStack cloud log data[13].

IV. OPENSTACK LOG ANALYSIS

In this section, it is used the OpenStack Log data collected, and the table shows the summary of Log data after applying the dimensionality reduction techniques Kernel PCA. The Table 2 describes the review of statistics of each variable.

TABLE 2: Summary of OpenStack Logs

	Component 1	*Component 2*
Minimum	-20.8800	-23.0914
1st Quadrant	-18.1100	-08.5381
Median	08.1700	-02.2703
Mean	00.0000	00.0000
3rd Quadrant	17.3100	-00.6545
Maximum	24.7300	28.4265

The Table 2 demonstrates that the range of values of component 1 is between -20.88 and 24.73 and the median is 8.17. The range of Component 2 is -23.0914 to 28.4265 with an average of -2.2703.

TABLE 3: Summary of Cluster Validation- OpenStack Logs

Clustering Methods:		*Hierarchical, K-Means, PAM*				
Cluster sizes:		2,3 ,4, 5, 6				
Validation Measures:		2	3	4	5	6
Hierarchical	Connectivity	0.0000	0.0000	2.7512	2.7512	7.7008
	Dunn	0.2873	0.2610	0.0400	0.0534	0.0387
	Silhouette	0.5717	0.8497	0.7935	0.8465	0.8282
K-Means	Connectivity	0.0000	5.2786	10.4516	2.7512	9.2984
	Dunn	0.2873	0.0191	0.0230	0.0534	0.0349
	Silhouette	0.5717	0.8555	0.8402	0.8465	0.8253
PAM	Connectivity	6.9603	5.6409	5.6409	5.6409	6.1643
	Dunn	0.0112	0.0192	0.0226	0.0242	0.0270
	Silhouette	0.6552	0.8555	0.8392	0.8156	0.8124

The Table 3 demonstrates the overall summary of performances of cluster validations on various clustering techniques at a different number of clusters.

Optimal Scores:

TABLE 4: Optimal Scores- OpenStack Logs

	Score	*Method*	*Clusters*
Connectivity	0.0000	Hierarchical	2
Dunn	0.2873	Hierarchical	2
Silhouette	0.8555	K-Means	3

The Table- 4 demonstrates the optimal scores obtained using the cluster validation techniques and based on the optimal ratings it can have the conclusion to go with Hierarchal Clustering on the OpenStack cloud log dataset.[14][15]

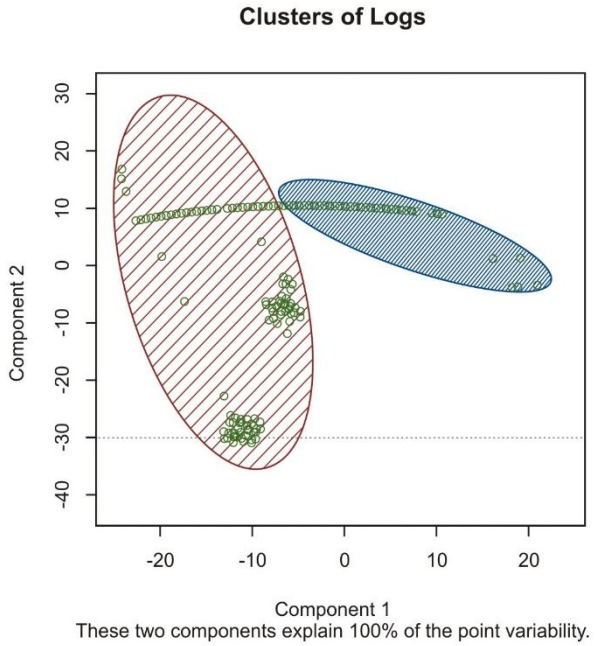

Fig- 2: Clusters of OpenStack Logs

The Fig-2 depicts the output of the hierarchal clustering as a plot. Here it sees a separate two individual clusters without any outliers[8][16]. Based on the cluster threshold system can easily categorize the clusters as separate cluster with normal behavior and cluster with abnormalities. Thus, now it can have the separate group of clusters formed from log dataset and can easily observe the characteristics of the bunch with defects[17].

V. HADOOP LOG ANALYSIS

In this section, it is used the Hadoop Log data collected, and the table shows the summary of Log data after applying the dimensionality reduction techniques Kernel PCA. The Table 5 describes the outline of the statistics for each variable. [18]

TABLE 5: Summary of Hadoop Logs

	Component 1	*Component 2*
Minimum	-30.9450	-23.4085
1st Quadrant	-28.0310	-11.9179
Median	07.1560	-00.6616
Mean	00.0000	00.0000
3rd Quadrant	18.5760	05.5925
Maximum	23.7540	26.3956

Logs

The Table 6 demonstrates the overall summary of performances of cluster validations on various clustering techniques at the different number of clusters.[19]

Clustering Methods:		*Hierarchical, K-Means, PAM*				
Cluster sizes:		2,3 ,4, 5, 6				
Validation Measures:		2	3	4	5	6
Hierarchical	Connectivity	0.0000	10.0151	10.7984	10.7984	10.7984
	Dunn	0.2618	0.0670	0.1002	0.1002	0.1002
	Silhouette	0.5586	0.6826	0.7634	0.8659	0.8341
K-Means	Connectivity	2.7107	18.9377	13.4028	16.5405	27.2948
	Dunn	0.0625	0.0244	0.1002	0.1002	0.0451
	Silhouette	0.5535	0.6822	0.7644	0.8684	0.8118
PAM	Connectivity	10.9861	14.8440	14.3738	16.5405	16.5405
	Dunn	0.0472	0.0660	0.1002	0.1002	0.1002
	Silhouette	0.4951	0.6832	0.7643	0.8684	0.8609

Table 6: Summary of Cluster Validation- Hadoop

Optimal Scores:

TABLE 7: Optimal Scores- Hadoop Logs

	Score	*Method*	*Clusters*
Connectivity	0.0000	Hierarchical	2
Dunn	0.2618	Hierarchical	2
Silhouette	0.8684	K-Means	5

The Table.7 demonstrates the optimal scores obtained using the cluster validation techniques and based on the optimal ratings, can have the conclusion to go with Hierarchal Clustering on the Hadoop log dataset.

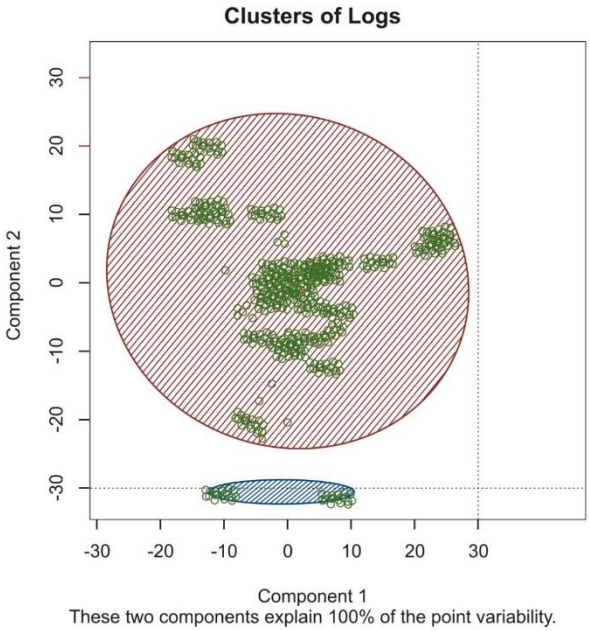

Fig.3 Clusters of Hadoop Logs

The Figur-3 depicts the output of the hierarchal clustering as a plot. Here it can see the separate two individual clusters without any outliers. Based on the cluster threshold system

can easily categorize the clusters as a cluster with normal behavior and cluster with abnormalities. Thus, now there can have the separate group of clusters formed from log dataset and can easily observe the characteristics of the bunch with defects. [20]

VI. SPARK LOG ANALYSIS& RESULTS

In this section, it is used the Spark Log data collected, and the table shows the summary of Log data after applying the dimensionality reduction techniques Kernel PCA. The Table 8 describes the outline of the summary for each variable.

TABLE 8: Summary of Spark Logs

	Component 1	*Component 2*
Minimum	-30.9450	-23.4085
1st Quadrant	-28.0310	-11.9179
Median	07.1560	-00.6616
Mean	00.0000	00.0000
3rd Quadrant	18.5760	05.5925
Maximum	23.7540	26.3956

Table 9: Summary of Spark Logs

Clustering Methods:		*Hierarchical, K-Means, PAM*				
Cluster sizes:		2,3 ,4, 5, 6				
Validation Measures:		2	3	4	5	6
Hierarchical	Connectivity	0.0000	0.3361	0.8151	3.9802	3.9802
	Dunn	0.1870	0.1679	0.1806	0.0474	0.0474
	Silhouette	0.5256	0.6086	0.6868	0.7114	0.7055
K-Means	Connectivity	10.6825	16.9468	14.1754	24.1988	20.9694
	Dunn	0.0151	0.0098	0.0213	0.0231	0.0242
	Silhouette	0.5353	0.6554	0.7086	0.7133	0.6949
PAM	Connectivity	15.7619	20.7464	16.5444	29.8151	40.7405
	Dunn	0.0017	0.0110	0.0235	0.0015	0.0017
	Silhouette	0.5347	0.6569	0.7069	0.7115	0.7290

The Table 9 demonstrates the overall summary of performances of cluster validations on various clustering techniques at the different number of clusters.

Optimal Scores:

	Score	Method	Clusters
Connectivity	0.000	Hierarchical	2
Dunn	0.187	Hierarchical	2
Silhouette	0.729	K-Means	6

TABLE 10: Optimal Scores- Spark Logs

The Table-10 demonstrates the optimal scores obtained using the cluster validation techniques and based on the optimal ratings, can have the conclusion to go with Hierarchal Clustering on the Spark log dataset[21].

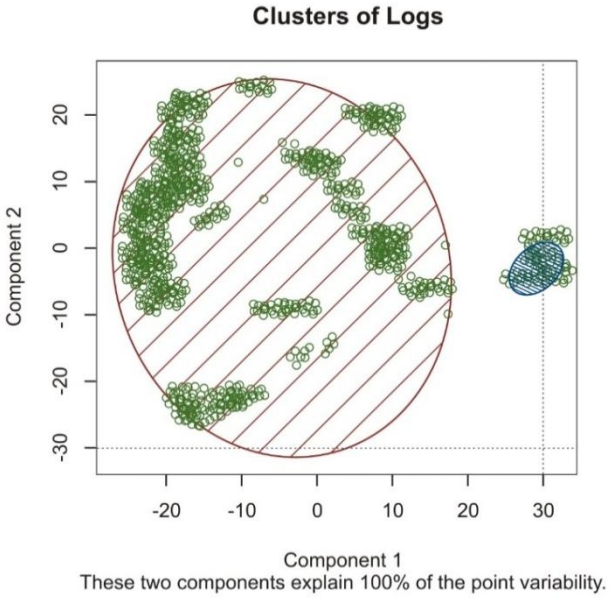

Fig 4: Clusters of Spark Logs

The Figure4 depicts the output of the hierarchal clustering as a plot. Here it can see a separate two individual clusters without any outliers. Based on the cluster threshold it can easily categorize the clusters as a cluster with normal behavior and cluster with abnormalities. [22]Thus, now it can have the separate group of clusters formed from log dataset and can easily observe the characteristics of the bunch with defects.[23]

VII. CONCLUSION

There is a need to have an effective mechanism to analyze the trillions of logs generated from various sources in every minute. The advantage of such models is it can identify or track the abnormal behavior of users in the environment without affecting the performances of the existing situation. Here it can see the clustering the new dataset will be automatically initiated based on the cluster validation techniques. It is impossible to analyze such a massive amount of data through rule-based engines, and the system envisaged a self-capable model of clustering using cluster validation techniques.

REFERENCES

1. P. Shi, “An Efficient Approach for Clustering Web Access Patterns from Web Logs,” vol. 5, pp. 1–14, 2009.
2. P. Dhanalakshmi and K. Ramani, “Clustering of users on web log data using Optimized CURE Clustering,” vol. 7, no. 5, pp. 2018–2024, 2018.
3. N. A. Subramaniam, “APPLICATION OF MACHINE LEARNING AND DEEP LEARNING ON NETWORK INTRUSION DETECTION.” 2017.
4. G. Di Modica and O. Tomarchio, “Matchmaking semantic security policies in heterogeneous clouds,” *Futur. Gener. Comput. Syst.*, vol. 55, pp. 176–185, 2016.
5. G. Suchacka, M. Skolimowska-kulig, and A. Potempa, “A k - Nearest Neighbors Method for Classifying User Sessions in E-Commerce Scenario,” no. 1, pp. 64–69.
6. L. Wang, “IoT Big Data Application Requirements,” 2015.

7. A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tutorials*, vol. PP, no. 99, p. 1, 2015.
8. T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Inf. Sci. (Ny).*, vol. 177, no. 18, pp. 3799–3821, 2007.
9. M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning," *ACM SIGSAC Ccs'17*, pp. 1285–1298, 2017.
10. R. Buyya, C. Shin, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms : Vision , hype , and reality for delivering computing as the 5th utility," *Futur. Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
11. M. C. Belavagi and B. Muniyal, "Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection," *Procedia Comput. Sci.*, vol. 89, pp. 117–123, 2016.
12. F. Wauthier, *Practical Machine Learning*. 2009.
13. "Test Dataset - Normal." [Online]. Available: https://www.cs.utah.edu/~mind/papers/deeplog_misc.html.
14. [14] M. T. Khorshed, A. B. M. Shawkat Ali, and S. A. Wasimi, "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing," *Futur. Gener. Comput. Syst.*, vol. 28, pp. 833–851, 2012.
15. P. M. El-kafrawy, A. A. Abdo, and A. F. Shawish, "Security Issues Over Some Cloud Models," *Procedia - Procedia Comput. Sci.*, vol. 65, no. Iccmit, pp. 853–858, 2015.
16. N. Paladi, C. Gehrmann, and A. Michalas, "Providing User Security Guarantees in Public Infrastructure Clouds," *IEEE Trans. Cloud Comput.*, vol. 5, no. 3, pp. 405–419, 2017.
17. R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning For Network Intrusion Detection," *2010 IEEE Symp. Secur. Priv. Outs.*, pp. 305–316, 2010.
18. J. Wei, Y. Zhao, K. Jiang, R. Xie, and Y. Jin, "Analysis farm: A cloud-based scalable aggregation and query platform for network log analysis," *Proc. - 2011 Int. Conf. Cloud Serv. Comput. CSC 2011*, pp. 354–359, 2011.
19. S. He, J. Zhu, P. He, and M. R. Lyu, "Experience Report : System Log Analysis for Anomaly Detection," 2016.
20. J. Zhao *et al.*, "A security framework in G-Hadoop for big data computing across distributed Cloud data centres," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 994–1007, 2014.
21. X. Lin, P. Wang, and B. Wu, "Log analysis in cloud computing environment with Hadoop and Spark," *Proc. 2013 5th IEEE Int. Conf. Broadband Netw. Multimed. Technol. IEEE IC-BNMT 2013*, pp. 273–276, 2013.
22. I. Mavridis and H. Karatza, "Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark," *J. Syst. Softw.*, vol. 125, pp. 133–151, 2017.
23. G. Adamson, M. Holm, P. Moore, and L. Wang, "A Cloud Service Control Approach for Distributed and Adaptive Equipment Control in Cloud Environments," *Procedia CIRP*, vol. 41, pp. 644–649, 2016.