***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Efficient Data Gathering in Cloud Computing Using Clustering Schemes, Cloud Agent-Based Data Schemes and Efficient Path Planning Techniques

Lalit Kumar[1]
[1]Research Scholar, Department of Computer Science & Engineering,
Shobhit Institute of Engineering & Technology, (NAAC Accredited Grade "A", Deemed to- be- University),
Meerut, India
lalit.chahal@gmail.com

Nidhi Tyagi[2]
[2]Professor, Department of Computer Science & Engineering,
Shobhit Institute of Engineering & Technology, (NAAC Accredited Grade "A", Deemed to- be- University),
Meerut, India
nidhi.tyagi@shobhituniversity.ac.in

Seema Gupta[3]
[3]Associate Professor, Ideal Institute of Management & technology,
GGSIP University,
Delhi, India
seemagupta.iimt@gmail.com

Abstract: Cloud computing is a rapidly evolving field with a wide range of applications, including data storage, processing, and analysis. Clustering schemes, cloud agent-based data schemes, and efficient path planning techniques are all important aspects of cloud computing. Clustering schemes are used to group similar data items together, which can improve the efficiency of data processing and analysis. Cloud agent-based data schemes allow for the collection and management of data from a variety of sources, including cloud and edge devices. Efficient path planning techniques are used to optimize the routing of data packets in cloud networks. This research paper provides a comprehensive overview of clustering schemes, cloud agent-based data schemes, and efficient path planning techniques for cloud computing. The paper discusses the different types of clustering schemes and cloud agent-based data schemes, as well as their advantages and disadvantages. The paper also presents an overview of efficient path planning techniques for cloud computing. The paper concludes by discussing the challenges and future research directions in cloud computing. The authors believe that cloud computing is a rapidly evolving field with a bright future, and they encourage researchers to continue to develop new and innovative solutions for cloud computing.

Keywords: cloud computing, clustering schemes, cloud agent-based data schemes, efficient path planning techniques

I. INTRODUCTION

In the ever-evolving landscape of cloud computing, the relentless pursuit of optimizing performance has become a cardinal objective for organizations and researchers alike. The burgeoning scale and complexity of cloud infrastructures necessitate innovative approaches to data management, resource allocation, and service provisioning. Clustering, as a fundamental technique, plays a pivotal role in harnessing the full potential of cloud environments, facilitating efficient data organization, resource utilization, and scalability. However, as the cloud computing ecosystem diversifies and matures, the need to comprehensively evaluate and compare different clustering schemes becomes increasingly imperative.

This research paper embarks on a journey into the heart of cloud computing, where a multitude of clustering paradigms, each endowed with unique characteristics, vie for prominence. Our investigation delves into the intricate nuances of these clustering schemes, shedding light on their individual strengths and weaknesses. The overarching goal of this study is to provide a comprehensive and unbiased analysis of how various clustering strategies perform in the dynamic and multifaceted arena of cloud computing.

In the pursuit of this objective, we will scrutinize a selection of diverse clustering schemes, ranging from traditional hierarchical and partition-based methods to cutting-edge density-based and spectral clustering techniques. By embracing a holistic perspective, we aim to not only quantify their performance but also uncover insights into their adaptability, scalability, and reliability within the context of cloud environments.

To underpin our analysis, we will employ an extensive set of metrics encompassing critical facets of cloud performance such as resource utilization, data distribution, and response times. Through rigorous experimentation and empirical evaluation, we will discern the strengths and weaknesses of each clustering scheme, providing valuable insights to guide cloud architects, data scientists, and system administrators in making informed decisions.

Ultimately, our research aspires to contribute to the ongoing dialogue surrounding the optimal utilization of cloud computing resources. By offering an unbiased assessment of the performance variability among different clustering schemes, we aim to empower cloud stakeholders with the knowledge and tools required to navigate the complex terrain of modern cloud environments effectively. As the cloud continues to shape the digital landscape, understanding the nuances of clustering in this context becomes not merely a choice, but a necessity for those striving to unlock the full potential of cloud computing.

II. CLUSTERING SCHEMES

Clustering in cloud computing is a crucial strategy that empowers cloud service providers to efficiently allocate computational resources and optimize performance. It involves the grouping of physical or virtual machines, servers, or nodes into clusters, each of which functions as a single entity for resource management and task allocation. Clustering schemes in cloud computing are diverse and multifaceted, designed to address the complex challenges posed by the dynamic and heterogeneous nature of cloud environments. This article delves into the intricacies of clustering schemes in cloud computing, highlighting their significance and exploring various types and their advantages. The significance of clustering in cloud computing cannot be overstated. Cloud environments are characterized by their vast scale, diverse workloads, and fluctuating demands. Clustering plays a pivotal role in resource allocation, load balancing, and fault tolerance, ensuring that cloud resources are utilized optimally. It enhances system performance, reliability, and scalability while minimizing energy consumption and operational costs. Clustering enables cloud providers to meet service-level agreements (SLAs) and maintain high levels of user satisfaction, making it an indispensable component of cloud infrastructure.

Clustering, within the realm of data analysis, emerges as a prominent optimization problem with the primary goal of partitioning datasets into meaningful groups or clusters. The process of clustering is multifaceted and can be executed through a variety of methodologies, each with its distinct characteristics. Two prevalent clustering paradigms are hard clustering and fuzzy clustering. In the context of hard clustering, the dataset is categorically divided into clusters, with each data point unequivocally belonging to a specific cluster. Conversely, in fuzzy clustering, a more nuanced approach is adopted, where each data point is associated with membership degrees for multiple clusters, indicating the level of affinity it holds for each cluster. The clustering activity encompasses a sequence of crucial steps, as depicted in Figure 1, which serves as a visual representation of the clustering process. This multifaceted process can be categorized into four overarching sections, each playing a pivotal role in the overall success of clustering endeavours:

- Clustering or Grouping: The initial phase involves the identification and grouping of similar data points into clusters based on chosen criteria or algorithms. This step sets the foundation for subsequent analyses.
- Clustering Algorithm Validation: Ensuring the efficacy of the chosen clustering algorithm is essential. This validation phase assesses the accuracy and appropriateness of the algorithm in generating meaningful clusters.
- Data Abstraction and Output Generation: Once clusters are established, the data is abstracted and summarized within each cluster. This abstraction results in a more concise representation of the dataset, facilitating subsequent analyses and insights. Additionally, the generation of output data, such as cluster centroids or representative exemplars, is crucial for further knowledge extraction.
- Knowledge or Meaningful Information: This phase involves the extraction of knowledge and meaningful information from the clustered data. It may entail statistical analyses, pattern recognition, or other techniques to uncover valuable insights within the clustered data.
- Pattern Representation and Proximity Measures: In the final section, the clustered patterns are represented in a manner that aids interpretation and decision-making. Proximity measures, such as distances between data points within clusters, are often employed to quantify relationships and similarities within and between clusters.

In summary, clustering is a fundamental data analysis technique with various approaches, including hard and fuzzy clustering. It involves a sequence of interrelated steps, ranging from initial data grouping to the extraction of valuable insights. The four core sections—clustering or grouping, clustering algorithm validation, data abstraction and output generation, knowledge extraction, and pattern representation and proximity measures—jointly contribute to the successful execution of clustering tasks, enabling data analysts and researchers to uncover hidden structures and patterns within complex datasets.

III. DIFFERENT CLUSTERING SCHEME

A diverse array of clustering algorithms exists, each serving specific purposes and exhibiting unique characteristics. These algorithms can be systematically categorized into distinct groups, providing a structured framework for understanding their underlying principles and applications [32]. Two major categories that encompass a wide range of clustering algorithms are:

A. ***Partitioning-Based Algorithms:*** This category includes renowned clustering methods such as K-means, K-medoids, Partitioning around Medoids (PAM), Clustering Large Applications (CLARA), and Clustering Large Applications based upon Randomized Search (CLARANS). Partitioning-based algorithms segment the dataset into non-overlapping clusters, where each data point belongs to only one cluster. The selection of an appropriate clustering technique within this category depends on factors like data distribution, cluster shape, and computational efficiency. The process of cluster formation is executed swiftly and effectively. It commences with the initial identification of clusters, which are subsequently adjusted and redistributed to establish associations. This process is predominantly achieved by partitioning datasets into a predefined number of data subsets, where each subset serves as a representation of a cluster. To establish a cluster, certain conditions must be met; specifically, every cluster must encompass at least one object, and every object must be affiliated with a cluster. The central concept underlying this approach revolves around the notion of a cluster center, often referred to as a centroid or medoid. Two prominent partitioning-based clustering methods, K-means and K-medoids, have gained significant recognition in this context. In the case of K-means, the cluster centers are iteratively adjusted to form new clusters until a state of convergence is achieved. Conversely, K-medoids represents an advancement of the K-means clustering approach, tailored to accommodate diverse datasets. It is worth noting that within this category of clustering algorithms, several alternatives exist, such as Partitioning around Medoids (PAM), Clustering Large Applications (CLARA), and Clustering for Arbitrary-shaped data with Noise (CLARANS). These methods have been engineered to exhibit relatively low time complexity and high efficiency, making them favourable choices for numerous clustering tasks. However, it is essential to acknowledge that this clustering paradigm is not suitable for non-convex data distributions, which are particularly sensitive to the determination of the number of clusters.

- K-means, a widely recognized squared error-based clustering algorithm, is instrumental in partitioning data into distinct clusters [28]. In this algorithm, 'k' signifies the number of clusters to be formed. However, to reach a conclusive solution, a stopping criterion is imperative, specifying the maximum number of allowable relocations. Each cluster is represented by a cluster head known as the centroid, and during each iteration, these centroids may shift, potentially leading to the formation of new clusters. Initially, the dataset is randomly divided into 'k' clusters based on certain pre-existing information.
- K-medoids, an extension of the K-means algorithm, seeks to minimize the sum of errors by selecting medoids as representatives of clusters. The primary goal is to find the configuration with the lowest cost. K-medoids demonstrates greater robustness against noise compared to K-means. The selection of medoids is contingent upon the minimal average dissimilarity of an object from the others within the cluster. The algorithm consists of two pivotal steps: the Build step, where 'k' centrally located objects are sequentially chosen as medoids, and the Swap step, in which the objective function is iteratively reduced by swapping selected and non-selected objects until no further reduction is possible.
- PAM (Partitioning Around Medoids), akin to K-medoids, centers its approach on medoid-based representation. PAM, much like K-medoids, aims to minimize dissimilarity between cluster representatives (medoids) and their respective members.
- CLARA (Clustering Large Applications) distinguishes itself through a focus on sampling techniques. To obtain a more accurate approximation for calculating medoids, CLARA employs multiple samples and subsequently selects the best clustering solution [37]. The quality of clustering is evaluated based on the average dissimilarity of all objects in the entire dataset, rendering more satisfactory results compared to PAM. It primarily operates on numerical data and exhibits a time complexity of O(Ks^2+K(N-K)). However, it is worth noting that CLARA does not handle high-dimensional data and lacks the capability to manage noisy data.
- CLARANS (Clustering Large Applications based on Random Search) introduces a different sampling approach compared to CLARA [30]. In CLARANS, a sample of neighbours is drawn at each step of the search process, contrasting with CLARA, where a sample of nodes is drawn at the outset of the search. This variance in sampling methods distinguishes CLARANS and contributes to its unique clustering approach.

B. ***Hierarchical-Based Algorithms:*** Hierarchical clustering techniques, including BIRCH, ROCK, and CURE, adopt a different approach. Rather than creating a set number of clusters from the outset, hierarchical algorithms build a tree-like structure or hierarchy of clusters, allowing for the exploration of clusters at multiple levels of granularity. Utilizing a proximity matrix is an integral step in the process of constructing a hierarchical structure from data, as discussed in reference [28]. The outcome of employing hierarchical clustering algorithms invariably yields a binary tree structure. In this hierarchical tree, the root node serves as a representation of the entire dataset, while the leaf nodes symbolize distinct subsets of the data. Intermediate nodes within the tree convey the degree of proximity between these subsets, and the height of the tree signifies the distance between each pair of data points within the cluster. It is important to note that hierarchical clustering techniques are not limited to a single level of clustering; rather, they facilitate the partitioning of the binary tree at various levels. Within this broader classification, hierarchical clustering can be further subdivided into two primary categories: agglomerative algorithms and divisive algorithms. Agglomerative algorithms are characterized by their bottom-up approach, wherein data points are initially treated as individual clusters and are successively merged together based on proximity, ultimately forming a hierarchical structure. Conversely, divisive algorithms take a top-down approach, commencing with the entire dataset as a single cluster and iteratively dividing it into smaller, more proximal clusters. These two categories represent distinct strategies within the realm of hierarchical clustering, each offering unique advantages and applications in the analysis of complex datasets.

- BIRCH (Balanced Iterative Reducing and Clustering Using Hierarchies):
  - BIRCH is a powerful unsupervised algorithm designed for hierarchical clustering in the context of handling extensive datasets [1], [28]. This algorithm possesses the remarkable capability to incrementally and dynamically create clusters from incoming multi-dimensional data objects, thereby producing optimal clustering results while efficiently managing available resources. A distinct advantage of BIRCH lies in its ability to accomplish this feat with just a single pass through the database, minimizing computational overhead. Furthermore, BIRCH is characterized by its compact representation, making it a resource-efficient choice for data clustering tasks. To achieve its hierarchical clustering objectives, BIRCH constructs a Clustering Feature (CF) tree incrementally, effectively organizing data objects into a hierarchical data structure.

- CURE (Clustering Using Representatives):
  - CURE is a proficient algorithm specially designed to address the challenges posed by large databases with varying cluster sizes and robustness concerns [37]. This algorithm's primary strength lies in its adaptability to handle datasets that exhibit substantial variations in cluster sizes, ensuring robustness in the clustering process.

- ROCK (Robust Clustering Using Links):
  - ROCK belongs to the category of agglomerative hierarchical clustering algorithms, as identified in reference [28]. To leverage the capabilities of the ROCK algorithm, users must install a CBA (Clustering by Approximation) package. ROCK enhances the quality of clustering outcomes through the strategic utilization of links between data points, optimizing the clustering process. However, it's essential to note that ROCK exhibits a higher time complexity of O(N^2*log N), which necessitates careful consideration when selecting it for clustering tasks in terms of computational resources and dataset size.

Understanding these distinct categories of clustering algorithms is essential for practitioners and researchers alike, as it aids in selecting the most appropriate technique for specific data analysis tasks. Depending on the nature of the data and the desired outcomes, one can choose either partitioning-based algorithms for straightforward, non-overlapping clustering or hierarchical-based algorithms for a more nuanced exploration of data relationships and structures.

IV. COMPARISON BETWEEN CLUSTERING SCHEME

Clustering Algorithm	Category	Shape of Data Set	Sensitive of Input Data	Sensitive to Outliers and Noise
K-means	Partition	Convex	High	High
K-medoid	Partition	Convex	Moderate	Little
PAM	Partition	Convex	Moderate	Little
CLARA	Partition	Convex	Moderate	Little
CLARANS	Partition	Convex	High	Little
BIRCH	Hierarchical	Convex	Moderate	Little
CURE	Hierarchical	Arbitrary	Moderate	Little
ROCK	Hierarchical	Arbitrary	Moderate	Little

V. PERFORMANCE COMPARISON BETWEEN DIFFERENT SCHEMES

Cloud computing has revolutionized the way we manage and deploy computing resources, making it crucial to optimize resource allocation and management. Clustering schemes in cloud computing have emerged as pivotal strategies to achieve these goals. In this section, we perform a comprehensive performance comparison between three prominent clustering schemes in cloud computing: BIRCH, CURE, and ROCK. Our analysis encompasses key performance metrics, including scalability, fault tolerance, resource optimization, and computational complexity.

A. Scalability:

- BIRCH: BIRCH demonstrates exceptional scalability due to its incremental and dynamic clustering approach. It can efficiently accommodate increasing workloads and datasets, making it an ideal choice for cloud environments that require on-the-fly adaptation to varying demands.
- CURE: CURE exhibits moderate scalability. While it can handle larger databases, its performance may degrade when faced with exceptionally extensive datasets. However, its adaptability to varying cluster sizes remains a strong point.
- ROCK: ROCK's scalability is limited due to its high computational complexity (O(N^2*log N)). It may struggle to scale gracefully in cloud environments with rapidly growing workloads or extensive datasets.

B. Fault Tolerance:

- BIRCH: BIRCH offers good fault tolerance, primarily due to its incremental nature. It can recover gracefully from node or cluster failures, ensuring minimal service disruption in cloud environments.
- CURE: CURE provides moderate fault tolerance. While it can handle some level of variation in cluster sizes, it

may struggle with more extensive and irregular fluctuations.

- ROCK: ROCK's fault tolerance is decent, but its high computational complexity may impact recovery times in the event of node or cluster failures.

C. Resource Optimization:

- BIRCH: BIRCH excels in resource optimization, particularly in terms of memory usage. Its compact representation and single-pass database scan contribute to efficient resource utilization.
- CURE: CURE's resource optimization is satisfactory. It efficiently manages resources in scenarios with varying cluster sizes, contributing to resource efficiency in the cloud.
- ROCK: ROCK's resource optimization is adequate, although its high time complexity can affect CPU and memory usage. Careful consideration is required when deploying ROCK in resource-constrained cloud environments.

D. Computational Complexity:

- BIRCH: BIRCH exhibits favourable computational complexity, making it suitable for real-time clustering tasks in cloud computing. Its single-pass approach results in a low time complexity, ensuring efficient processing.
- CURE: CURE's computational complexity is reasonable, offering a balance between scalability and resource efficiency. It remains a viable choice for cloud environments with moderately sized datasets.
- ROCK: ROCK's high computational complexity (O(N^2*log N)) is a notable limitation. It may pose challenges in resource-constrained cloud environments and should be employed judiciously.

VI. CONCLUSION

In the ever-evolving landscape of cloud computing, the choice of clustering scheme emerges as a critical factor influencing the efficiency, reliability, and scalability of cloud-based systems. This article has explored several prominent clustering schemes within the realm of cloud computing, each designed to address specific challenges and optimize resource utilization. As we draw our conclusions, it becomes evident that the diverse array of clustering schemes caters to the multifaceted needs of cloud environments, offering distinct advantages and trade-offs.

Firstly, BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) stands out as an unsupervised algorithm capable of hierarchical clustering for extensive datasets. Its ability to incrementally form clusters, while requiring only a single database scan, highlights its resource efficiency. The compact representation it provides makes it a compelling choice for certain cloud applications. CURE (Clustering Using Representatives), on the other hand, excels in robustly handling large databases with varying cluster sizes. Its adaptability to these challenges ensures that it remains a valuable tool for cloud practitioners dealing with diverse and dynamic datasets. ROCK (Robust Clustering using Links), while enhancing the quality of clustering through strategic link utilization, does come with the caveat of higher time complexity. This factor necessitates careful consideration when choosing it for specific cloud clustering tasks, especially in scenarios where computational resources and dataset sizes are significant.

In conclusion, the selection of a clustering scheme in cloud computing should be a well-informed decision, considering the specific requirements of the cloud environment and the goals of resource optimization. BIRCH, CURE, and ROCK represent just a few of the diverse options available, each tailored to different aspects of clustering in the cloud. Ultimately, the success of cloud-based applications and services relies on the judicious selection and effective implementation of these clustering schemes, enabling cloud providers to deliver the reliability, scalability, and efficiency that modern users and businesses demand in the ever-advancing world of cloud computing.

REFERENCES

[1] Rui Xu and Donald Wunsch., "Survey of clustering algorithms", IEEE Transactions on neural networks, Vol-16, Issue- 3, pp. 645-678, 2005.
[2] Banerjee, A., Pati, B. and Panigrahi, C.R, "SC2 : A SelectionBased Consensus Clustering Approach", Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2016, 2, p.177, 2017.
[3] Yang, Haibo, and Mary Tate, "A descriptive literature review and classification of cloud computing research", CAIS 31, 2, 2012.
[4] Ambroise, Christophe, Geniève Sèze, Fouad Badran, and Sylvie Thiria, "Hierarchical clustering of self-organizing maps for cloud classification", Neurocomputing 30, no. 1-4, pp. 47-52, 2000.
[5] Weinhardt, Christof, Arun Anandasivam, Benjamin Blau, Nikolay Borissov, Thomas Meinl, Wibke Michalk, and Jochen Stößer, "Cloud computing–a classification, business models, and research directions", Business & Information Systems Engineering, Vol-1, no. 5, pp.391-399, 2009.
[6] Swagatika Shrabanee, Rath Amiya Kumar, Pattnaik Prasant Kumar, Markov Chain Model And PSO Technique for Dynamic Heuristic Resource Scheduling for System Level Optimization of Cloud Resources, ARPN Journal of Engineering and Applied Sciences. 13(7): 375-393, 2018.
[7] Fritzke, Bernd. "Growing cell structures—a self-organizing network for unsupervised and supervised learning", Neural networks, Vol-7, no. 9, pp.1441-1460, 1994.
[8] Sindhwani, Vikas, Partha Niyogi, and Mikhail Belkin, "Beyond the point cloud: from transductive to semi-supervised learning", In Proceedings of the 22nd international conference on Machine learning, pp. 824-831. ACM, 2005.
[9] Andrea Baraldi and Ethem Alpaydin, "Constructive feed forward ART clustering networks", IEEE transactions on neural networks 13, no. 3, pp. 645-661, 2002.
[10] Panda, Prasanta Kumar, and Swagatika Shrabanee. "Energy consumption in cloud computing and power management", International Journal of Advanced Research in Computer Science, Vol- 3, no. 2, 2012.
[11] Bernd Fritzke, "Some competitive learning methods", Artificial Intelligence Institute, Dresden University of Technology , 1997.
[12] Zhang, Qingchen, Laurence T. Yang, Zhikui Chen, and Peng Li, "PPHOPCM: privacy-preserving high-order possibilistic

c-means algorithm for big data clustering with cloud computing", IEEE Transactions on Big Data, 2017.
[13] Chris Fraley, and Adrian E. Raftery, "MCLUST: Software for model-based cluster analysis", Journal of classification 16, no. 2, pp.297-306, 1999.
[14] Anil K. Jain, Robert P. W. Duin, and Jianchang Mao, "Statistical pattern recognition: A review", IEEE Transactions on pattern analysis and machine intelligence, Vol-22, no. 1, pp.4-37, 2000.
[15] Gu, Guofei, Roberto Perdisci, Junjie Zhang, and Wenke Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocoland Structure-Independent Botnet Detection", In USENIX security symposium, vol. 5, no. 2, pp. 139-154. 2008.
[16] Ryszard S. Michalski, and Robert E. Stepp, "Automated construction of classifications: Conceptual clustering versus numerical taxonomy",IEEE Transactions on pattern analysis and machine intelligence,pp.396-410,1983.
[17] Victor L. Brailovsky, "A probabilistic approach to clustering", Pattern Recognition Letters 12, no. 4, pp.193-198, 1991.
[18] D.Singh,B. Pattanayak,C. Panda, "Analysis of an Improved Energy Balanced Routing Protocol for Wireless Sensor Network", IEEE International conference on Communication and Signal processing(ICCSP- 2016),pp-1503-1507, 2016.
[19] John H. Holland, "Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence", Ann Arbor, MI: University of Michigan Press, pp. 439-444, 1975.
[20] Jianchang Mao, and Anil K. Jain, "Texture classification and segmentation using multi resolution simultaneous autoregressive models", Pattern recognition 25, no. 2, pp.173-188, 1992.
[21] Mahendiran, A., N. Saravanan, N. Venkata Subramanian, and N. Sairam, "Implementation of K-means clustering in cloud computing environment", Research Journal of Applied Sciences, Engineering and Technology, Vol- 4, no. 10, pp.1391-1394, 2012.
[22] Karen L. Oehler, and Robert M. Gray, "Combining image compression and classification using vector quantization", IEEE transactions on pattern analysis and machine intelligence 17, no. 5, pp. 461-473, 1995.
[23] Berkhin, Pavel, " A survey of clustering data mining technique", In grouping multidimensional data, pp. 25-71. Springer, Berlin, Heidelberg, 2006.
[24] Fionn Murtagh, "A survey of recent advances in hierarchical clustering algorithms", The Computer Journal 26, no. 4, pp. 354- 359, 1983.
[25] Andrea Baraldi, and Palma Blonda, "A survey of fuzzy clustering algorithms for pattern recognition", IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 29, no. 6, pp.778-785, 1999.
[26] Swagatika, Shrabanee, and Pattnaik P. K., "Design Criteria of SOA for Cloud Based Infrastructure Resource Management as a Service", International Journal of Instrumentation, Control & Automation (IJICA),Vol- 1, no-1 ,2011.
[27] A.Rauber, J. Paralic, and E. Pampalk, "Empirical evaluation of clustering algorithms," J. Inf. Org. Sci., vol. 24, no. 2, pp. 195- 209, 2000.
[28] Chih-Ping Wei, Yen-Hsien Lee, and Che-Ming Hsu, "Empirical comparison of fast clustering algorithms for large data sets", In System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, pp. 10-pp. IEEE, 2000.
[29] Scheunders, Paul, "A comparison of clustering algorithms applied to color image quantization", Pattern Recognition Letters, Vol-18, no. 11-13, pp.1379-1384, 1997.
[30] Boomija, M. D., and M. Phil, "Comparison of partition based clustering algorithms", Journal of Computer Applications, Vol-1, no. 4, pp.18-21, 2008.
[31] Ahmad, Amir, and Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data", Data & Knowledge Engineering, Vol-63, no. 2, pp.503-527, 2007.
[32] D. Singh, B.K. Pattanayak, "Analytical Study of an Improved Cluster based Routing Protocol in Wireless Sensor Network", Indian Journal of Science and Technology, Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/97947, October 2016.