Turkish Online Journal of Qualitative Inquiry (TOJQI)

Volume 12, Issue 3, June 2021: 2286-2297

Research Article

EFFECTIVE METRICS, DATA CLUSTERING AND SEARCHING MECHANISMS FOR DATA MANAGEMENT IN ERP SYSTEMS

K. Mohammed Hussain [1], J. Thangakumar [2], D. Venkata Subramanian[3]

Abstract

This paper proposes a fuzzy based clustering search approach for an ERP system and searching mechanisms along with most appropriate metrics. In real time scenarios it is cumbersome to search for a particular piece of data from a data mart used by ERP applications. The situation becomes worse when it is difficult to search for the string when numerous strings are available. This work proposes a clustering approach which uses a fuzzy inference engine to make the search more effective and fast. The proposed technique reduces the processing time by 82% and the memory usage by 8%, compared to the conventional technique of searching the students' or faculty data from the experimental ERP data sets. This paper also provides the comprehensive list of the metrics which can be considered to be incorporated as part of the framework and deployment of any ERP system.

Keywords:Levenshtein distance, Fuzzy inference engine, Data Clustering, ERP, Processing time, Memory Consumption, Metrics, Searchability, Clustering

I. INTRODUCTION

In today's world, an ERP system is of immense importance in several industrial sectors, namely, educational, health, finance, etc. It becomes important to have an effective and elegant strategy to manage the databases in such conditions. The data needs to be updated, precise and presentable in a particular form to extract useful information and take appropriate decisions in a timely manner. ERP systems search using very large data sets. The databases are used for storing useful information pertaining to ERP processes.

[1]Research Scholar, Department of Computer Applications, Hindustan Institute of Technology & Science, Chennai, India

[2]Associate Professor, Department of Computer Science, Hindustan Institute of Technology & Science, Chennai, India

[3]Department of Computer Science, Velammal Institute of Technology, Chennai, India

For example, the ERP data sets include student's data, faculty data, admin data, payment data, etc. This work primarily focuses on searching the related students' or faculty data which are used further in other ERP processes. The proposed search results would enhance the data migration, data integration and pre-processing. The precise and matching records are selected in a timely manner and make processing faster. In several cases, the data may not be complete; it may be partial or incomplete. Several algorithms related to data search based on Levenshtein distance of data points are explained in [1, 2, 3, 4, and 5]. In such cases, there would be an adverse impact on the database search engines.

The situation would be worse when multiple ERPs are being integrated in a cloud or Big Data environment, where databases are being spanned irrespective of the locality of reference of the particular data item of interest. In the circumstances cited, the most critical parameter of interest would be the processing time and memory storage. Many data mining algorithms require advanced computations and storage mechanisms to complete the processing efficiently. In this work, it would be ideal to organise and group the data in a clustered manner. This could be achieved by making a dependency analysis on the data available. In this work, a clustering algorithm is used to group the data sets. We attempt to model the data as points and store them in small clusters. Each cluster has a parent and a group of children. The clusters themselves can be modelled as a decision tree, using the parent -child relationship.

II. LITERATURE REVIEW

Table 1. List of references and Issues Addressed

Reference No.	Author	Paper title	Issue addressed
1	Michael Elkin and Seth Patie	A linear-size logarithmic Stretch Path-Reporting distance Oracle for General Graphs	Levenshtein distance between query points
2	Andris Ambainis, Wiliam Gasarch, Aravind Srinivasan and Andrey Utis	Lower bounds on the Deterministic and Quantum communication complexity of Hamming-Distance Problems	Query bounds on distance between query points
3	Pauli Miettinen and Jilles Vreeken	MDL4BMF: Minimum Description Length for Boolean Matrix Factorisation	Query length optimisation
4	Chuan Lei and Elke A. Rundensteiner	Robust Distributed Query Processing for Streaming Data	Query performance
5	Lu-An Tang, Yu Zheng and Jing Yuan , Jiawei Han, Alice Leung, Wen-Chih Peng, Thomas La Porta	A framework of Travelling Companion Discovery on Trajectory Data Streams	Processing multiple streams of data simultaneously
6	Jonathan A.Silva, Elaine R.Faria, Rodrigo C.Barris, Eduardo R.Hruschka,	Data Stream Clustering: A Survey	Clustering based approach creating models for data objects

	Andre C.P.L.F De Carvalho and Joao Gama.		
7	Lan Cao and Hongwei Zhu.	Normal Accidents: Data Quality Problems in ERP- Enabled Manufacturing	ERP Data Quality Aspects, Normal Accident theory in data quality aspects
8	Levi Shaul and Doron Tauber.	Critical Success factors in Enterprise Resource Planning Systems: Reviews of the Last Decade	The vital factors that need to be addressed for the successful implementation of ERP
9	Anton Rytting, David Zagic	Spelling Correction for Dialecal Arabic Dictionary Lookup	Spell Check errors in dictionary lookup for queries
10	Kostas Kolomvatsos, Christos Anagnostopoulos and Stathes Hadjiefthymiades.	A Fuzzy Logic System for Bargaining in Information Markets	Fuzzy based logic for web inference. Artificial Inteligence for learning aspects.
11	Leonid Boytsov	Indexing Methods for Approximate Dictionary Searching : Comparative Analysis	Indexing techniques for faster search options in dictionary lookup
12	Rafail Ostrovsky and Yuval Rabani	Low Distortion Embeddings for Edit Distance	Edit distance metric in Levenshtein distance concepts
13	Surrendra Baswana and Sandeep Sen	Approximate Distance Oracles for Unweighted Graphs in Expected $O(n^2)$ Time	Graph based approach to formulate least distances
14	Mikkel Thorup and Uri Zwick	Approximate Distance Oracles	Approximation on Oracle based distances
15	Gonzalo Navarro	A Guided Tour to Approximate String Matching	Edit distances metric on string length and associated costs
16	Andrea Bonarini and Gianluca Bontempi	A Qualitative Simulation Approach for Fuzzy Dynamical Models	Quasi Qualitative approach on Fuzzy sets.
17	Michael Wolfe	The Definition of Dependence Distance	Concept of dependence distance.
18	William Pugh	Definitions of Dependence Distance	Normalisations' benefits of dependence distances of query points.
19	Pradheep Kumar.K. And Venkata Subramanian. D.	Fuzzy-Based Querying Approach for	Fuzzy Based Query approach in SQL

		Multidimensional Big Data Quality Assessment	
20	D.Venkata Subramanian and K. Pradheep Kumar	Fuzzy Based Modeling for an effective IT Security Policy Management	Fuzzy based Rule set for decision making

In order to extract useful information, the raw data required for the information may be available in multiple databases as explained in [6, 7, 8, 9, and 10]. Several indexing methods and data clustering methods reported in literature have been discussed in [11, 12, 13, 14, and 15]. The data points distance computation based on noise elimination has also been explained in [17, 18]. The databases are mapped as data sets which may be intersected among themselves. We then choose different queries to obtain the information. The choice of the query set used for processing depends on the locality of reference of the data. For instance, if the data required is available in the same data set, the data could be easily retrieved and processed. Hence, the query processing time would be the least. To accomplish the same a Fair query set could be used. On the contrary, if the data belongs to two immediate clusters of close proximity, the proximity factor is determined by the query point distance which could be computed by the Euclidean distance formula. If we have a query point (x_p,y_p,z_p) and 2 data points, one from cluster 1 given by (x_1,y_1,z_1) and another from cluster 2 given by (x_2,y_2,z_2), the distances are given by the equations

$$d1=\left(\sqrt{(x_p-x_1)^2+(y_p-y_1)^2+(z_p-z_1)^2}\right) \text{----------------------- (1)}$$

and

$$d2=\left(\sqrt{(x_p-x_2)^2+(y_p-y_2)^2+(z_p-z_2)^2}\right) \text{----------------------- (2)}$$

Depending on the values of d1 and d2, the proximity of the clusters could be determined. If the clusters are of lesser proximity, we could use an optimal query set. If the proximity value is very high we could use a Quick processing set, as discussed by Pradheep and Venkat in [19]. The Fuzzy based modelling by constructing rule sets for decision making has also been illustrated by Venkat and Pradheep in [20]. The references and the issue addressed by them have been tabulated in Table 1.

III. FUZZY BASED SEARCHING

In this section the design methodology has been explained. The entire data mart is logically split into a number of clusters. Each cluster has a parent and a group of child nodes. The given query string that needs to be searched is accepted as input and 3 parameters are computed:

- Start match (Start)
- Min match
- Max match
- Worst match

The Start match parameter is computed by taking a single character of the string. The min match parameter is computed by taking the least possible string length of the characters required to find a match from the cluster. The max match parameter is computed by taking the average string length of the characters required to make the search optimistic or promising. The worst match parameter is computed by taking the entire string length of the characters required to make the search precise and accurate. Data points are formulated by taking a triplet as follows (min, max and worst). The fuzzy based matching is carried out by fixing a threshold. The threshold would decide on the number of exact matches that would be returned. In real time scenarios it may not be always possible that the query string and the data string? Precisely match and such searching mechanisms would take a very long time. The fuzzy based approach first sets an arbitrary threshold for the entire data mart for the query string to be located. The matching is carried out against the parent nodes of the task clusters. It then identifies the clusters which match with similar thresholds. It then computes an average of the thresholds and searches for clusters that have thresholds greater than the computed average threshold. The process continues till we identify only one cluster. After identifying the appropriate cluster, the search is carried out against the child nodes to precisely identify the string.

The proposed methodology and the workflow have been discussed below:

- Partition the entire data mart into logical clusters of an ideal size
- The data points are mapped into the cluster groups each with a parent node.
    - Clusters to be grouped as parents with relevant child nodes
    - Compute parameters min match, max match and worst match
- Accept the query string
- Set an arbitrary threshold
- If the result set has n matching clusters
    - Compute the average of the thresholds
    - Restrict the search to parents equal and above the average thresholds
    - Repeat the process till we are zero down to the particular cluster
        - Make comparison against all child nodes under this cluster
- End the procedure.

The workflow has been illustrated by a flowchart as shown in Fig 1.

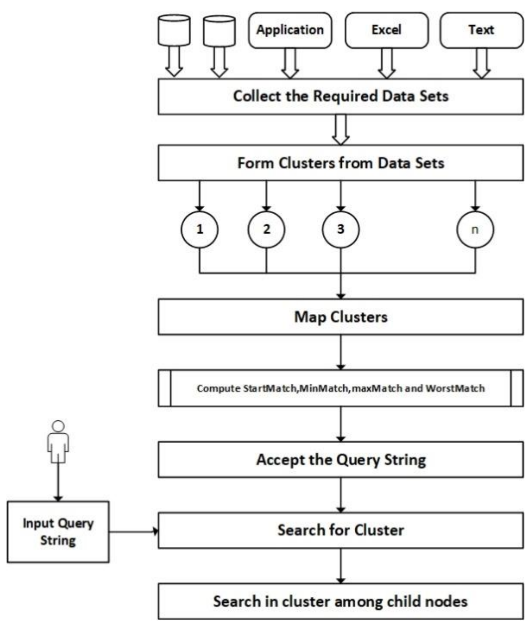

Fig 1. Flowchart illustrating workflow

IV. EXPERIMENTAL EVALUATION

The ERP system Data Mart chosen to simulate the algorithm contains 10,255 records of students from a university. It contains 795 clusters with 795 parent nodes. Each cluster has 20 child nodes. The algorithm has been simulated using Microsoft Visual Studio 2010. The ER diagram showing the attributes of each cluster is given in Figure 3. The cluster table showing different attributes is also given below. The different string clusters are grouped and illustrated in Fig 2. The subsets of the required strings of the data sets are processed by the algorithm to extract useful information.

Fig 2. Clusters formulated from ERP data-mart

An ER diagram showing the different attributes to simulate the algorithm is illustrated in Fig 3.

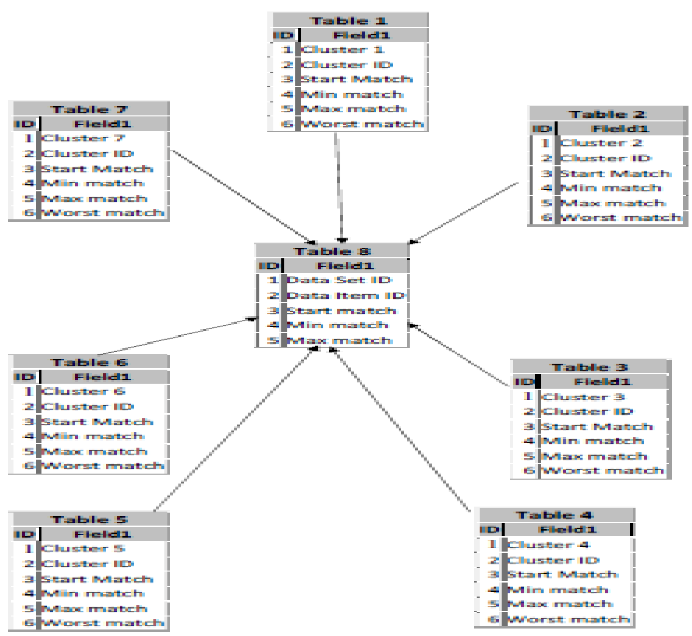

Fig 3. ER diagram for Cluster Connectivity

V. METRICS AND RESULTS

The overall success of ERP system and its components can be evaluated and measured using the following key metrics.

(i). Accuracy – to measure the how accurate the data about the student, courses, faculty and others

(ii). Consistency – to measure the variation in the data formats and consistency of data between different departments and components.

(iii). Velocity – to measure on how quickly the data changes flows through the entire system and its components.

(iv). Infrastructure Cost – to measure the cost spent on the overall infrastructure

(v). Load Cycle Time – to measure the length of time it takes to load the batch data and real time data to the ERP systems and the related databases.

(vi). Demand Forecast Accuracy – to measure how accurately predict the future demand in terms of the user data to allocate additional resources and capacity to deal with the scalability

(vii). Schedule Adherence – to measure how accurately the ERP systems and components relates to schedule of adhering to time to load, process and produce the reports

(viii). Downtime – to measure how long the ERP and its associated components were not available to ensure the minimal down time

(ix). Service Availability – to measure the availability of the required ERP service to support a particular functionality and/or operation.

(x). Search Time – to measure the time taken to search the given data in the ERP systems

(xi). Processing Time – to measure the processing time taken for fetching and processing the given information

(xii). Memory Usage – to measure the memory consumed in terms of KB or MB for any given transaction or unit of work in the ERP system

(xiii). CIA – to measure the confidential access, integrity of the data and availability of security 24*7

Among all the above metrics, memory usage and processing time are most important ones as they directly contribute to the performance of the ERP system. This work identifies that the performance results of the same which is shown in Table 2. It could be observed that with an increase in the number of data sets there is a reduction of processing time in the clustering approach. The average reduction in memory consumed is about 81.46 % as indicated in the table. This occurs for around 5000 data sets.

Table 2. Processing time - Data Mart Search Vs Clustering Approach

S.No.	Number of data sets	Processing time		Reduction in Processing time (%) R= ((D-C)/C)*100)
		Data Mart (D)	Clustering (C)	
1	1000	0.154	0.02	90.24
2	2000	0.105	0.02	84.47
3	3000	0.084	0.02	70.95
4	4000	0.097	0.02	80.29

5	5000	0.110	0.02	81.60
6	6000	0.121	0.02	82.76
7	7000	0.056	0.01	77.47
8	8000	0.059	0.01	81.98
9	9000	0.111	0.02	83.40
Average		0.10	0.02	81.46

A plot showing the comparison of the processing times of both the approaches is shown in Fig 4. It could be observed that for 1000 data sets the processing time is about 0.159ms in the data sets approach, wheres the processing time is only 0.019 ms for the same number of data sets.

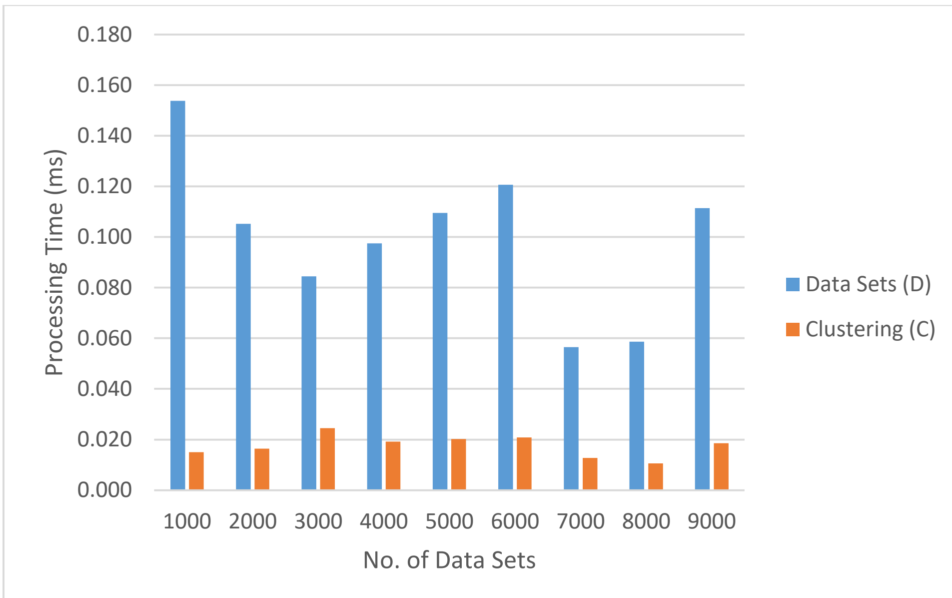

Fig 4. Plot Comparing Processing times

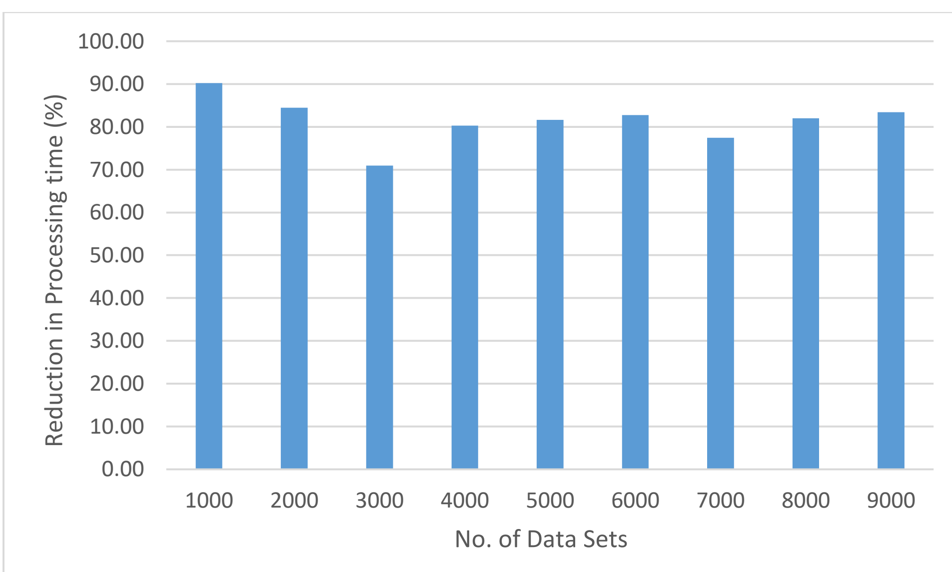

Fig 5. Plot - Reduction in processing time

It could be observed from Fig 5 that a reduction in processing time of 90.27% has been obtained for about 1000 data sets. We get the average reduction of 81.46% for about 2000 data sets. The results of the memory utilisation are shown in Table 3. The average reduction in memory consumed is about 7.72 % as indicated below.

Table 3. Comparison of memory utilisation: Data Sets Approach Vs Clustering Approach

S.No.	Number of data sets	Memory consumed		Reduction in Memory consumed (%) R= ((D-C)/C)*100)
		Data Sets (D) (Mbytes)	Clustering (C) (Mbytes)	
1	1000	57.84	55.63	3.82
2	2000	60.49	55.63	8.03
3	3000	60.51	55.63	8.06
4	4000	61.05	55.63	8.88
5	5000	60.69	55.63	8.33
6	6000	60.53	55.63	8.09
7	7000	60.53	55.63	8.09
8	8000	60.53	55.63	8.09
9	9000	60.53	55.63	8.09
Average		55.63	55.63	7.72

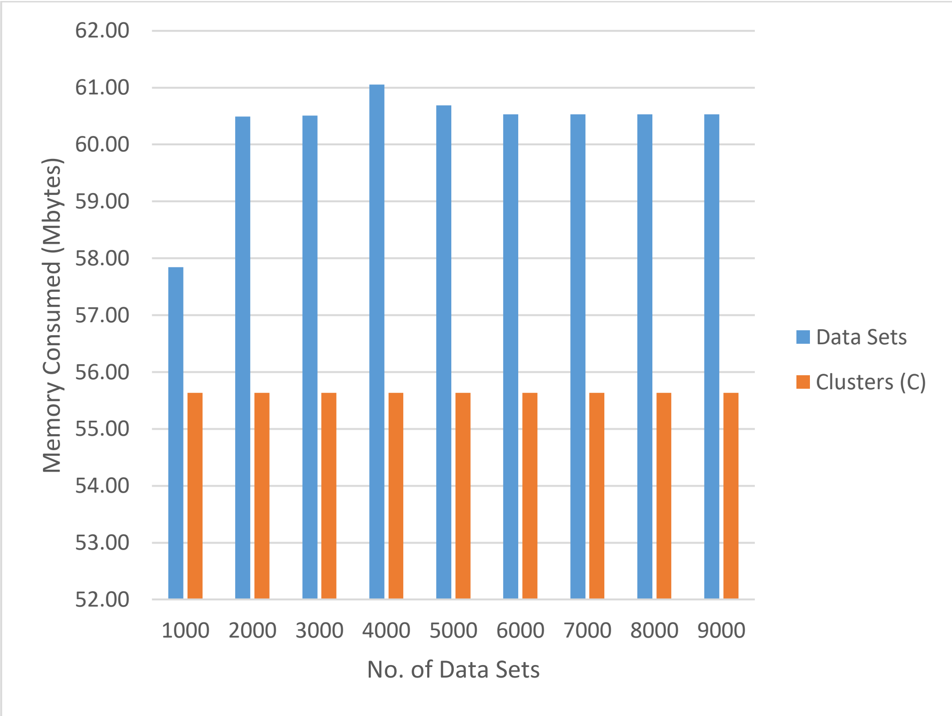

Fig 6. Plot Comparing Memory Utilisation

A plot comparing the memory utilisation of both the approaches (Data Sets Vs Clustering Approach) has been shown in Fig 6. It could be observed that the maximum memory utilisation for 4000 data sets is about 61 Mbytes on using the Data Sets approach, whereas the maximum memory utilisation is only about 55.9 Mbytes, on using the clustering approach.

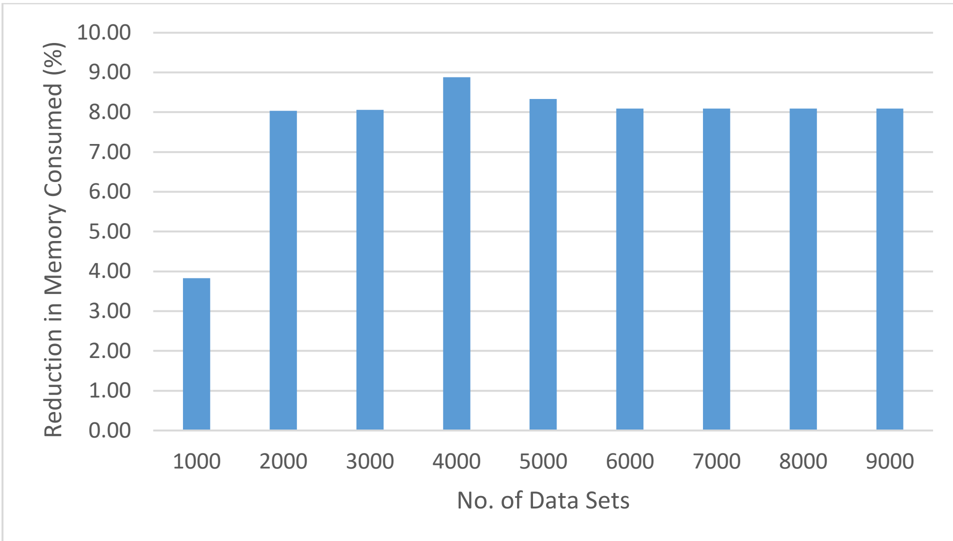

Fig 7. Plot - Reduction in memory consumption

A plot illustrating the reduction in memory consumption of both the approaches has been shown in Fig 7. The maximum reduction of 8.9% has been obtained for about 4000 data sets on using the clustering approach. Corresponding to a drastic reduction of processing time which is nearly about 82%, we get a memory reduction of about 8%. This is because the clustering approach makes additional search iterations in storing tentative search results. When the search is carried out directly from the data sets formulated, the memory utilisation would be the same but the processing time would be very large.

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, a cluster based search approach has been implemented in the traditional ERP system dataset as a standalone application. In this approach, the processing time reduces to a very large extent. One of the future directions would be to implementing more robust secured layer using IAM which will ensure both trust and confidentiality of the ERP data access. The challenges that would be involved would be of security and scalability. These challenges could be overcome by using efficient security based algorithms.

References

[1] Michael Elkin and Seth Pettie. "A Linear-Size Logarithmic Stretch Path-Reporting Distance Oracle for General Graphs", pp 50:1-50:31, ACM Transactions on Algorithms, Vol.12, No.4, Article 50, pp 50:1-50:30, Aug 2016

[2] Andris Ambainis, William Gasarch, Aravind Srinivasan and Andrey Utis. "Lower Bounds on the Deterministic and Quantum Communication Complexity of Hamming-Distance Problems", ACM Transactions on Computation Theory, Vol7, No.3, Article 10, pp10:1-10:10, Jun 2015.
[3] Pauli Miettinen and Jilles Vreeken "MDL4BMF: Minimum Description Length for Boolean Matrix Factorisation", 2014 ACM 1556-4681/2014/10-ARTS18, ACM Transactions on Knowledge Discovery from Data, Vol.8, No.4, Article 18, pp 18:1-18:31, Oct 2014.
[4] Chuan Lei and Elke A. Rundensteiner. "Robust Distributed Query Processing for Streaming Data", 2014 ACM 0362-5915/2014/05-ART17, ACM Transactions on Database Systems, Vol.39, No.2, Article 17, pp17:1-17:44,May 2014.
[5] Lu-An Tang, Yu Zheng and Jing Yuan , Jiawei Han, Alice Leung, Wen-Chih Peng, Thomas La Porta. "A framework of Travelling Companion Discovery on Trajectory Data Streams", 2013 ACM 2157-6904/2013/12-ARTS, ACM Transactions on Intelligent Systems and Technology, Vol 5, No.1, Article 3, pp3:1-3:33, Dec 2013.
[6] Jonathan A.Silva, Elaine R.Faria, Rodrigo C.Barris, Eduardo R.Hruschka, Andre C.P.L.F De Carvalho and Joao Gama. "Data Stream Clustering: A Survey", 2013 ACM 0360-0300/2013,10-ARTS13, ACM Computing Surveys, Vol 46, No.1, Article 13, pp13:1-13:30, Oct 2013.
[7] Lan Cao and Hongwei Zhu. "Normal Accidents: Data Quality Problems in ERP- Enabled Manufacturing", 2013 ACM 1936-1955/2013/05-ART11, ACM Journal of Data and Information Quality, Vol.4 No.3, Article 11, pp 11:1-11:26,May 2013
[8] Levi Shaul and Doron Tauber. "Critical Success factors in Enterprise Resource Planning Systems: Reviews of the Last Decade", 2013 ACM 0360-0300/2013/ 08-ART55, ACM Computing Surveys, Vol 45, No.4, Article 55, pp 55:1 – 55:39 , Aug 2013.
[9] Anton Rytting, David Zagic, Paul Rodrigues, Sarah Wayland, Christian Hettick, Tim Buckwalter and Charles Blake. "Spelling Correction for Dialecal Arabic Dictionary Lookup"2011 ACM 1530-0226/2011/03-ART3, ACM Transactions on Asian Language Information Processing, Vol 10, No.1,Article 3, pp 3:1-3:15, Mar 2011.
[10] Kostas Kolomvatsos, Christos Anagnostopoulos and Stathes Hadjiefthymiades. " A Fuzzy Logic System for Bargaining in Information Markets", 2012 ACM 2157-6904/ 2012/02-ART32, ACM Transactions on Intelligent Systems and Technology, Vol 3, No.2, Article 32, pp 32:1-32:26,Feb 2012.
[11] Leonid Boytsov. "Indexing Methods for Approximate Dictionary Searching : Comparative Analysis", ACM Journal of Experimental Algorithmics, Vol 16, No.1, Article 1, pp 1:1-1:89,May 2011.
[12] RafailOstrovsky and Yuval Rabani. "Low Distortion Embeddings for Edit Distance", Journal of the ACM, Vol 54, No.5, Article 23, pp 23:1-23:16, Oct 2007.
[13] SurrendraBaswana and Sandeep Sen. " Approximate Distance Oracles for Unweighted Graphs in Expected $O(n^2)$ Time", ACM Transactions on Algorithms, Vol 2, No.4, pp 557-577,Oct 2006.
[14] MikkelThorup and Uri Zwick. "Approximate Distance Oracles", Journal of the ACM, Vol 52, No.1, pp 1-24, Jan 2005.
[15] Gonzalo Navarro. "A Guided Tour to Approximate String Matching", 2001 ACM 0360-0300/01/0300-0031, ACM Computing Surveys, Vol.33, No.1, pp 31-88, Mar 2001.

[16] Andrea Bonarini and GianlucaBontempi. “A Qualitative Simulation Approach for Fuzzy Dynamical Models”, 1994 ACM 1049-3301/94/1000-0285, ACM Transactions on Modelling and Computer Simulation, Vol 4, No.4, pp 285-313, Oct 1994.
[17] Michael Wolfe “The Definition of Dependence Distance”, ACM 0164-0925/94/0700-1114, ACM Transactions on Programming Languages and Systems”, Vol 16, No.7, pp 1114-1116, Jul 1994.
[18] William Pugh “Definitions of Dependence Distance”, ACM Letters on Programming Languages and Systems, Vol 1, No.63, pp 261-265, Sep 1992.
[19] Pradheep Kumar.K. and VenkataSubramanian.D. “Fuzzy-Based Querying Approach for Multidimensional Big Data Quality Assessment”, IGI Book Chapter for Book titled “Handbook of Research on Fuzzy and Rough Set Theory in Organisational Decision Making”, pp1-23, ISSN: 2327-3429; eISSN: 2327-3437), IGI, 2015.
[20] D.Venkata Subramanian and K. Pradheep Kumar. “Fuzzy Based Modeling for an effective IT Security Policy Management”, 978-1-4673-8460-5/16,pp 173-181, IEEE,2016