Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

INFORMATION COMPRESSION OF BIG DATA USING THE SP THEORY OF INTELLIGENCE

Vishmitha[1], Prof. B. S. Umashankar[2]

[1]IV Semester, M.Tech, Computer Science and Engineering, Sahyadri College of Engineering and Management
[2]Professor, Department of Computer Science and Engineering, Sahyadri College of Engineering and Management

Abstract—Big data includes large quantity of data that results drawbacks both in accessing and managing the data. These drawbacks are overcome by the introduction of the SP theory of Intelligence. The term 'SP' indicates 'Simplicity' and 'Power'. The central theme used in the theory is lossless information compression. This helps in making the big data small, thereby provides benefits both in accessing and management. The purpose of this project is to overcome the problems in big data using the SP Theory of Intelligence. In order to achieve this goal, big data is subjected to clustering and compression techniques. Compression of information is achieved by pattern matching. Using such a system leads to the improvement in the processing of big data. The SP Theory provides pattern recognition, information storage, retrieval and information compression. Although this theory leads in faster information retrieval, the integrity of the original information is maintained. Future work has to be done in this area to work with patterns in two dimensions.

Keywords—data clustering; data compression; pattern recognition; access time; original search; compressed search.

I. INTRODUCTION

Big data includes large volumes of data [1] ranging from Petabyte to Exabyte. This imposes several technical challenges including:

1. Scalability: Big data contains variety of data from distributed locations that poses difficulty both in its collection and integration.
2. Storage and Management: It is difficult to store and manage the collected heterogeneous datasets and thereby provide fast information retrieval.
3. Big data analytics should mine massive data in real-time or near real-time.

These challenges in big data are overcome by applying various compression techniques as shown in Figure 1. The three techniques of information compression [2] are:

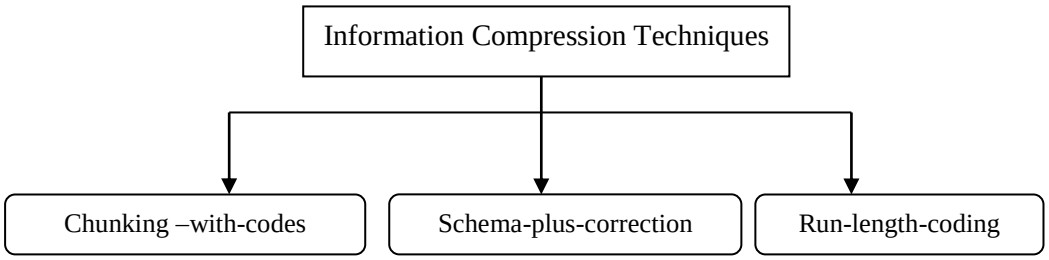

Figure 1. Techniques used in information compression

1. Chunking-with-codes

When a set of statements repeats in two or more parts of the computer program then it is better to declare those statements as a 'function', 'procedure' or 'sub-routine' within the program. Further each sequence can be replaced with a 'call' to the function. Here, the function is called as a 'chunk' and name of the function as 'code'.

Example: International Journal of Engineering is given a short name IJE.

2. Schema-plus-correction

It is similar to chunking-with-codes but with corrections or variations in the unified chunk of information in various occasions.

Example: A particular menu in any restaurant may have the form as: 'Menu 1: Appetizer (S) sorbet (M) (P) coffee-and-mints'. The choices are marked as 'S' for Starter, 'M' for Main course and 'P' for Pudding. Then any meal can be encoded as 'Menu1: (3) (5) (1)'. Here the digits indicate the choices of starter, main course and pudding.

3. Run-length-coding

It is used when there exists more than one copy of a pattern, each one except the first immediately following after its predecessor.

Example: Multiplication is repeated addition (3*4 = 3+3+3+3).

Application of any of these compression techniques results in compression of body of information 'I' which consists of two parts:

1. Grammar: contains patterns that appear repeatedly in I.
2. Encoding: contains non-redundant information in I.

The proposed approach overcomes the limitations of big data, that is, it reduces the access time of big data by applying compression technique called schema-plus-correction. Here an existing database is taken into account and a detailed study is conducted. The compression technique is applied to the existing database to reduce its volume. Finally a comparison is made by accessing a particular record from database before applying IC and after applying IC and the access time is noted down. It is observed that the time to access or search for the required record from the database after applying IC is greatly reduced. This approach can be employed in large enterprise organizations involving huge databases to reduce the complexity not only in storage but also in management [3] and transmission.

II. RELATED WORK

J. Gerald Wolff [1] has proposed the SP system to overcome the problem of variety in big data. Central in the system is lossless compression of information, which makes the big data smaller and thus reduces the problems in storage and management. Because of the concept of information compression via the matching and unification of patterns, the system may be applied to the representation and processing of all kinds of knowledge. SP theory finds for full and partial matches between the patterns. In contrast to unsupervised learning, SP theory compresses information (I) and creates grammar (G) and encoding (E). Depending on I and G interpretation achieves pattern recognition, retrieval of information, representation from one representation to another, planning and problem solving. Han Hu, Yonggang Wen, Tat Seng Chua and Xuelong [2] have proposed a Map Reduce framework for massive data analysis which consists of a single master Job Tracker and one slave Task Tracker. The master is responsible for scheduling jobs for slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master. The Map Reduce framework and Hadoop Distributed File System (HDFS) run on the same set of nodes that allows tasks to be on the nodes in which data is already present.

III. PROPOSED SYSTEM

3.1. System Architecture

The system architecture is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communication between these components. The 'Admin' will login. An existing database is taken into account that contains over twenty-three-lakh records. This database can be called as the 'Initial' or 'Original database'. This database contains Sensor information of a room. It includes 54 SensorIds. Each of the SensorId contains thousands of records. Initially a search is made for any one of the SensorId from the 'Original database'. We call this search as 'Original Database search'. Once the required records are retrieved, the time taken to retrieve those records is noted down.

In the next step, the 'Original database' is subjected to 'Data Clustering' as shown in Figure 2. The clustering process is done by pattern recognition [4] that is, by Matching and Unification of Patterns (MUP). It is found that in the database the patterns repeating are SensorIds, so we cluster based on the SensorIds. Since there are 54 SensorIds, we cluster them into nine clusters each cluster with six SensorIds namely ClassA, ClassB, ClassC, ClassD, ClassE, ClassF, ClassG, ClassH, ClassI.

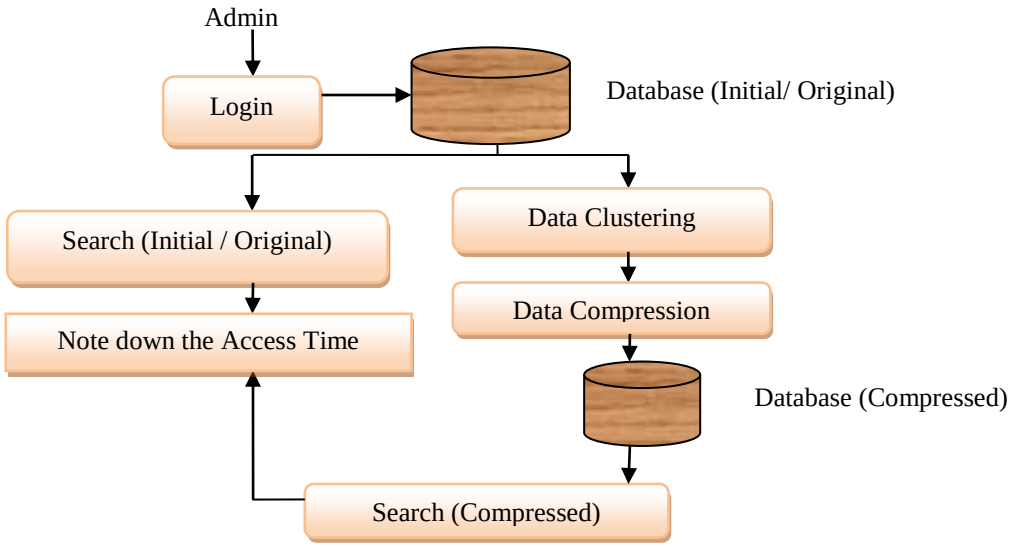

Figure 2. System Architecture

Finally the clustered data is to be compressed. Here the SensorIds along with the cluster to which they belong and their MasterIds are compressed and stored. This will help this will help to search for a particular record from the compressed MasterIds rather than from the entire database. At this stage a new database with both clustered and compressed data is obtained which is called ' Compressed database.' Now a search is made for the same SensorId as searched in initial search from the 'Compressed database'. We call this search as 'Compressed Database search'. Once the required records are retrieved, the time taken to retrieve those records is noted down. It is found that the time to retrieve a record from 'Compressed database' is much less than that of 'Original database'. This indicates lossless information compression is achieved and information retrieval is faster. This helps in the efficient management of big data [5]. After performing these operations the Admin logs out.

3.2. Decomposition Description

A pictorial representation of an algorithm is termed as a flow chart which shows the steps in boxes of various kinds, and arrows relate the order in which they connect as shown in Figure 3. Flowcharts are mainly used for analyzing, designing, or managing a process or program in various fields. In the proposed approach a search for a particular record of interest is done before and after applying

compression techniques. Finally a time comparison is conducted to discover that the time taken to access a certain record after the application of compression is greatly reduced.

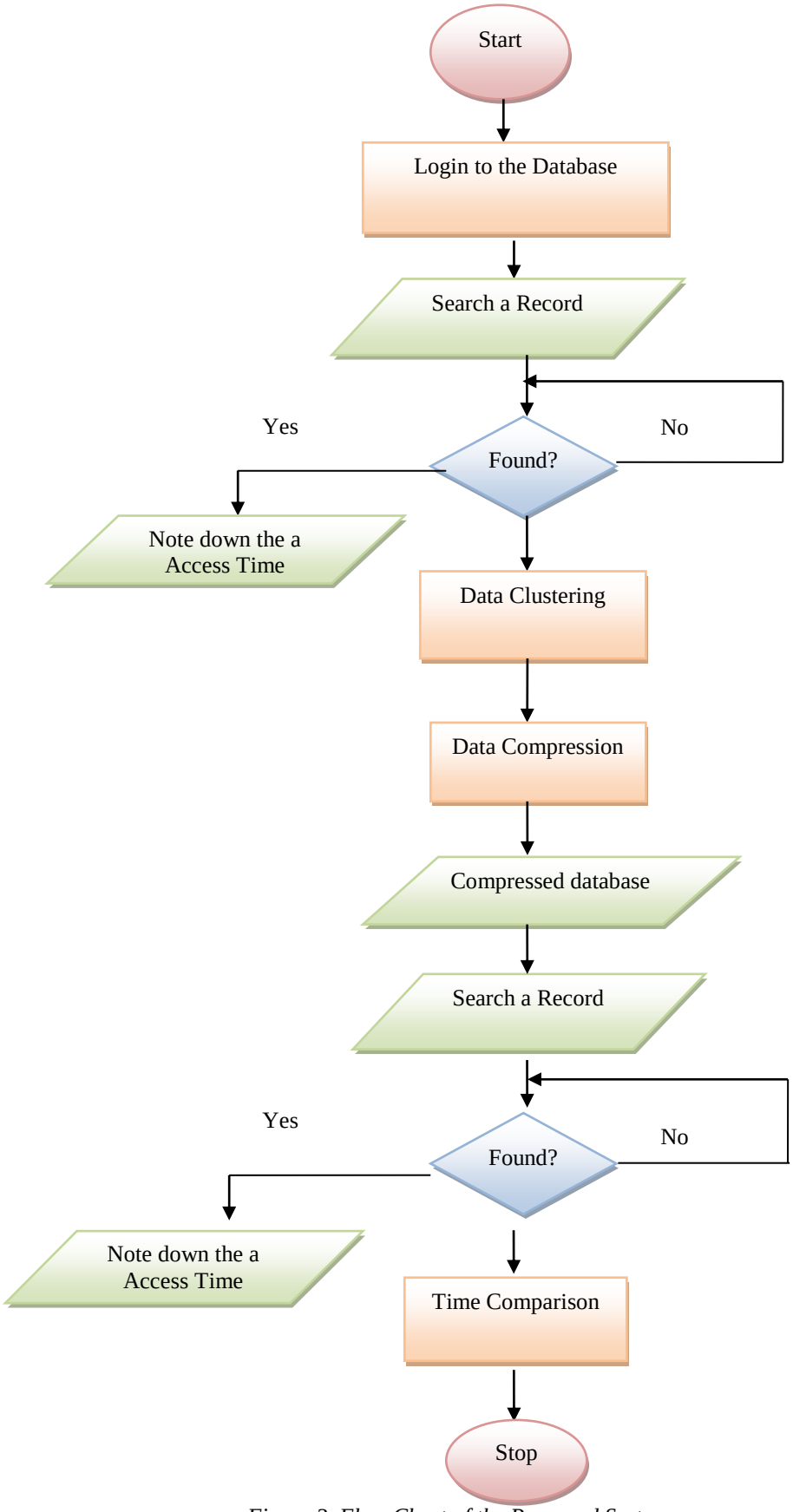

Figure 3. Flow Chart of the Proposed System

IV. IMPLEMENTATION RESULTS

The implementation includes six modules namely; Login, Original Search, Data Clustering, Data Compression, Compressed search and Comparison graph. Once the Admin enters a valid Username and Password, he can log in. He enters into an interface where he is allowed to perform five operations namely; searching (original), clustering, compression, searching (compressed) and comparison graph as shown in Figure 4.

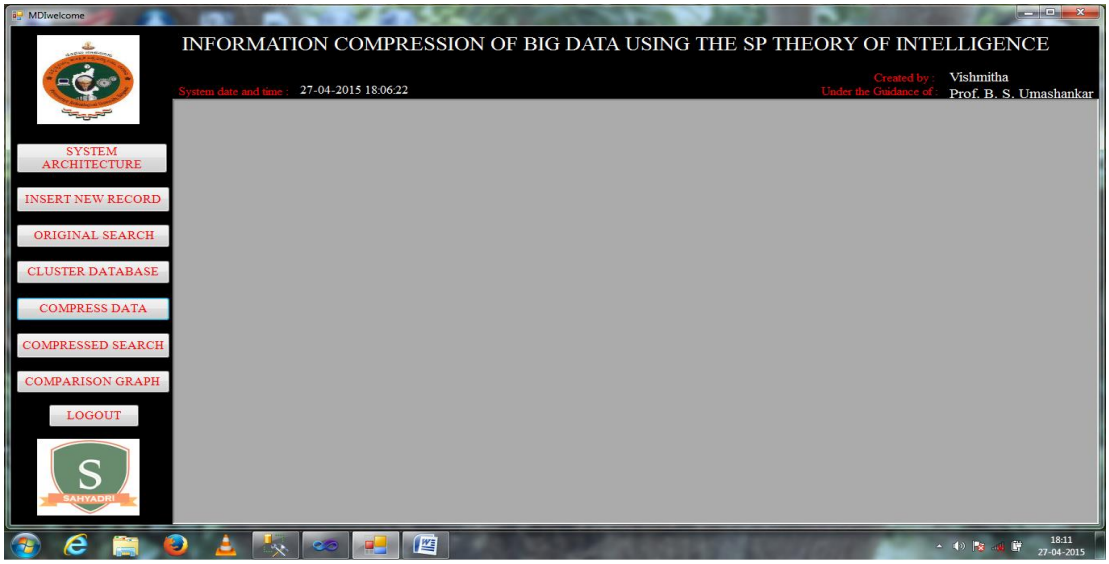

Figure 4. Interface to perform various operations on the Database

4.1.Original Search Module

Here the existing database is taken into account without making any modifications to it. The Admin searches for any record of interest based on the SensorId's. Once the required record is retrieved, the time taken to search that particular record is noted down. Time is calculated in millisecond as shown in Figure 5. The Access Time to retrieve records with SensorId 5 is 4294milliseconds.

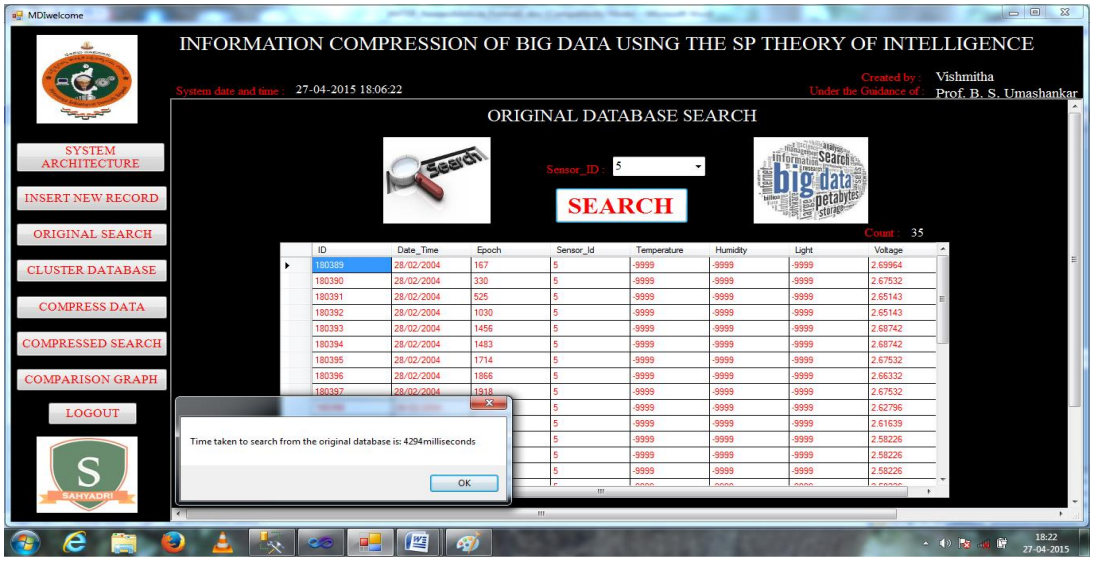

Figure 5. Original Search

4.2. Data Clustering Module

In this module the Admin performs updating the database, which means each record in the database is assigned a 'Class'. The database is initially not clustered. The clustering process is done by pattern recognition, that is, by Matching and Unification of Patterns (MUP) [6]. It is found that in the database taken into account, the patterns repeating are SensorIds, so we cluster based on the SensorIds as shown in Figure 6. In the Figure, SensorId 5 is clustered as ClassA. Since there are 54 SensorIds, we cluster them into nine clusters each cluster with six SensorIds namely ClassA, ClassB, ClassC, ClassD, ClassE, ClassF, ClassG, ClassH and ClassI.

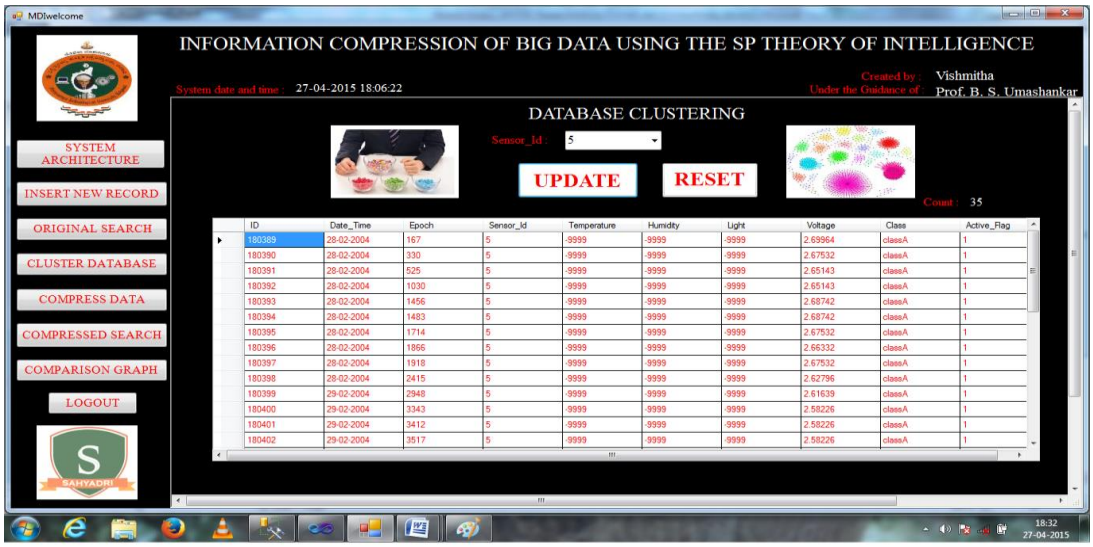

Figure 6. Data Clustering

4.3. Data Compression Module

The fourth operation is to compress the database. This is done by saving the MasterIDs of records in the database along with their respective SensorId's and Class in one more table. Here the SensorId's along with the cluster to which they belong and their MasterIDs which are unique for each record are stored in a separate table as shown in the Figure 7. This will help this will help to search for a particular record from the compressed MasterIDs rather than from the entire database.

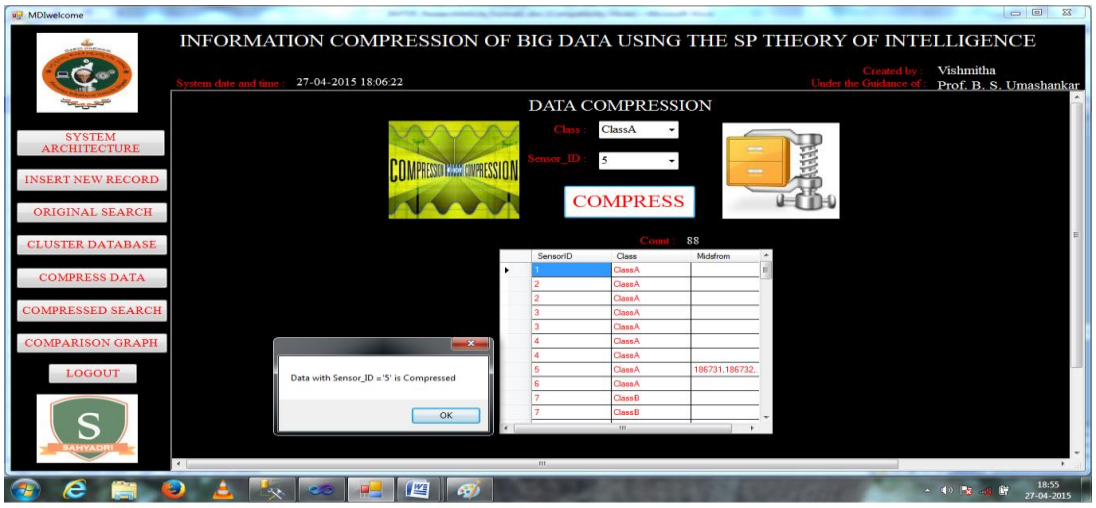

Figure 7. Data Compression

4.4. Compressed Search Module

Here a search is made from the compressed database i.e., after allotting the Classes and the time to search is noted down as shown in the Figure 8. The Access Time to retrieve records with SensorId 5 after compression is 16milliseconds. It is found that the time to search a record from the compressed database takes very less time than that from the original database. Since the information retrieval is faster in the compressed database, it indicates that it helps to reduce complexities in storage and lends itself to faster processing and transmission of data.

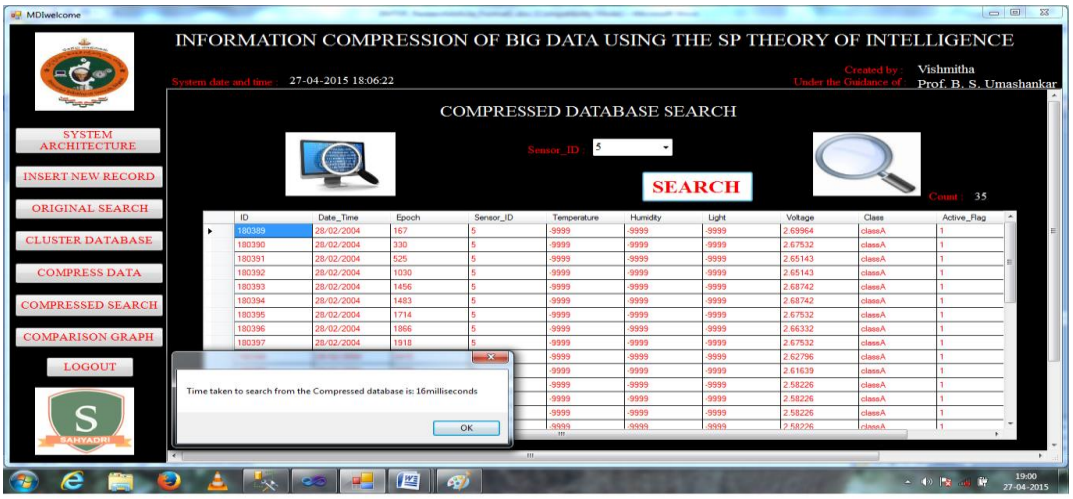

Figure 8. Compressed Search

4.5. Comparison Graph Module

The graph in Figure 9 shows a comparison of two searches namely, the 'Original Search' and the 'Compressed Search'. The comparison clearly shows that the 'Compressed Search' takes very less access time as compared to that of the 'Original Search'. Similar searches are done for SensorId's 17 and 24 and the results are shown in the graph. After performing these operations the user logs out from the database.

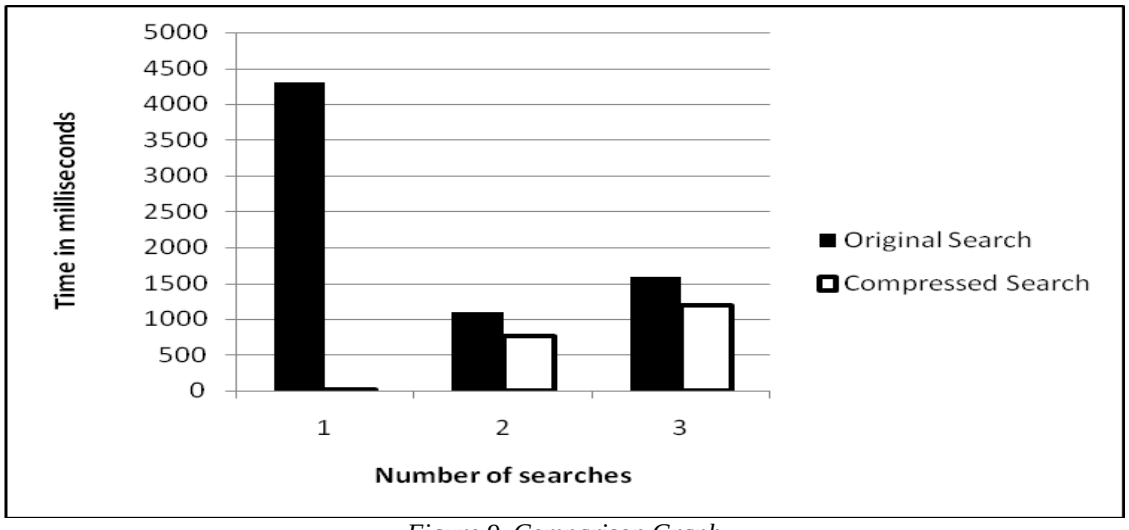

Figure 9. Comparison Graph

V. CONCLUSION

Data Clustering performed based on pattern recognition and lossless information compression applied to the clustered data makes the information retrieval faster and reduces the access time in big data thereby provides benefits in processing and management of the big data. The reduction in the access time does not affect the integrity of the original data.

REFERENCES

[1] J.G. Wolff, "Big data and SP theory of Intelligence", vol. 2, pp. 301-315, 2014.
[2] Han Hu, Yonggang Wen, Tat-Seng Chua and Xuelong Li, "Towards Scalable Systems for Big Data Analytics", vol. 6, no. 1, 2013.
[3] J.G. Wolff, "Big data and the SP theory of Intelligence", vol. 2, pp. 301-315, 2014.
[4] J. G. Wolff, "The SP theory of intelligence: An overview," Information, vol. 4, no.3, pp. 283-341, 2013.
[5] J. G. Wolff, "The SP Theory of intelligence: Benefits and Applications," Information, vol. 5, no. 1, pp. 1-27, 2014.
[6] J. G. Wolff, "The SP Theory of Intelligence: An Introduction", 2014.

