*Impact Factor (SJIF): 3.632*
**International Journal of Advance Research in Engineering, Science & Technology**
*e-ISSN: 2393-9877, p-ISSN: 2394-2444*
***Volume 3, Issue 6, July-2016***

# Data Mining Itemset Parallelization and Distribution Using Mapreduce Approach

[1] Ms.Shreedevi C Patil , [2]Prof.B.N.Veerappa B.E,M.TECH

[1]*P.G.Student*

[2]*ASSOCIATE PROFESSOR*

*Department of studies in Computer science and engineering*

*UBDT College of Engineering, Davanagere*

***Abstract-Existing parallel mining algorithms for frequent itemsets unavailable for the mechanism that renders automatic parallelization, load balancing, data distribution, and fault tolerance on large clusters. As a solution to this problem, we build a parallel frequent itemsets mining algorithm called FiDoop using the MapReduce programming model. To achieve compressed storage and keep away from Sbuilding conditional pattern bases, FiDoop introduce the frequent items ultrametric tree, rather than conventional FP trees. In FiDoop, three MapReduce jobs are implemented to complete the mining task. In the importance of third MapReduce job, the mappers independently decompose itemsets, the reducers perform combination operations by constructing small ultrametric trees, and the actual mining of these trees separately. We implement FiDoop on our in-house Hadoop cluster. We prove that FiDoop on the cluster is sensitive to information distribution and dimensions, because itemsets with distinct lengths have distinct decomposition and construction costs. To improve FiDoop's performance, we develop a workload balance metric to measure load balance across the cluster's computing nodes. We develop FiDoop-HD, an extension of FiDoop, to speed up the mining performance for high-dimensional data analysis.***

***Keywords—Frequent itemsets, frequent items ultrametric tree (FIU-tree), Hadoop cluster, load balance, MapReduce.***

## I.INTRODUCTION

### 1.1 Overview

Hadoop is known as an open-source system that allows us to store and process tremendous information in a conveyed situation where it is extended over bunches of PCs employing straightforward programming models. Here intention of hadoop is to scale up from single servers to a huge number of machines, every offering neighborhood calculation and capacity.

Because of the approach of new advances, gadgets, and correspondence implies like informal communication destinations, the amount of information delivered by humankind is becoming quickly consistently. The amount of information delivered by us from the earliest starting point of time till 2003 was 5 billion gigabytes. On the off chance that you heap up the information as plates it might fill a whole football field. The same sum was made in at regular intervals in 2011, and in like clockwork in 2013. This rate is as yet becoming gigantically. In spite of the fact that this data delivered is important and can be valuable when handled, it is being dismissed.

Huge information implies truly a major information, it is considered as an extensive datasets that can't be handled utilizing customary processing strategies. Enormous information is not simply an information, rather it has turned into a complete subject, which includes different instruments, methods and structures. Enormous data includes the data delivered by various gadgets and applications. Given underneath are a portion of the fields that go under the umbrella of Big Data.

Discovery Data : It is a part of helicopter, planes, and flies, and so forth. It receives voices of the flight group, recordings of amplifiers and headphones, and the execution data of the air ship.

Online networking Data : Social media, for example, Facebook and Twitter hold data and the perspectives posted by a huge number of individuals over the globe.

Stock Exchange Data : The stock trade information holds data about the "purchase" and "offer" choices made on an offer of various organizations made by the clients.

Power Grid Data : The force lattice information holds data devoured by a specific hub regarding a base station.

Transport Data : Transport information incorporates model, limit, separation and accessibility of a vehicle.

Web index Data : Search motors recover bunches of information from various databases.

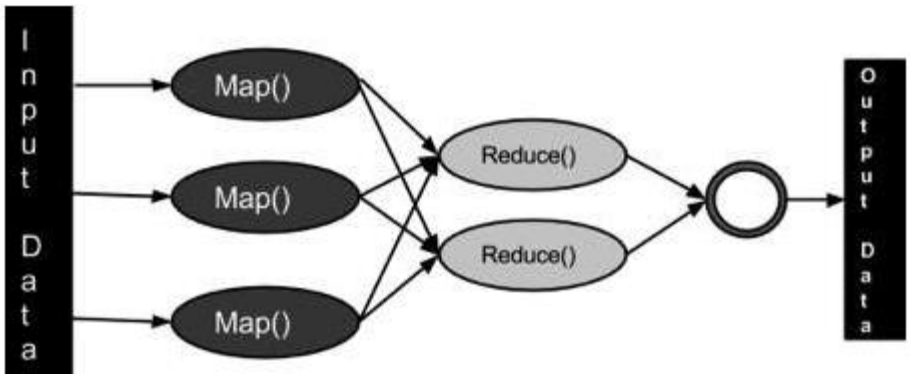


***Fig 1: Hadoop Prospective***

**Overview on Chronic Disease**

The chronic disease is a group of referential symptom disease based on the factor of age. For example, polio is affected for a human is only the age of 5 and less and similar the nitrification issues are noticed under the age group of 35 to 50 and their=r by many more. Each medical imparity comes with respect to the time and hence this approach is proposed to simulate and redefine the system behavioral model for analysis under these syndrome diseases.Today the medical parametics treatment has increased and the economic condition has come down for affording such diseases treatment. Hence from out project and mining application is proposed for designing and understanding the system requirement under mining.

## II.LITERATURE SURVEY

Frequent itemsets mining (FIM) is a core problem in association rule mining (ARM), sequence mining, and the like. Speeding up the process of FIM is critical and indispensable, because FIM consumption accounts for a significant portion of mining time due to its high computation and input/output (I/O) intensity. At the point when datasets in cutting edge information mining applications turn out to be too much huge, successive FIM calculations running on a solitary machine experience the ill effects of execution disintegration. To address this issue, we explore how to perform FIM utilizing MapReduce—a generally received programming model for handling enormous datasets by abusing the parallelism among figuring hubs of a group. We demonstrate to convey a substantial dataset over the bunch to adjust load over all group hubs, along these lines improving the execution of parallel FIM.

### 2.1 Survey Analysis

The paper summaries the following contribution

1) We made a complete overhaul to FIUT (i.e., the frequent items ultrametric trees method), and addressed the performance issues of parallelizing FIUT.
2) We developed the parallel frequent itemsets mining method (i.e., FiDoop) using the MapReduce programming model.

3) We proposed a data distribution scheme to balance load among computing nodes in a cluster.
4) We further optimized the performance of FiDoop and reduced running time of processing high-dimensional datasets.
5) We conducted extensive experiments using a wide range of synthetic and real-world datasets, and prove fidoop is more effective and efficient.

**2.2 FIU Tree Study**

The FIUT approach adopts the FIU Tree for enhancement of the efficiency of mining &constructed as follows.

1)After the root is named as invalid, an itemset (p1, p2, . . . , pm)of incessant things is embedded as a way associated by edges (p1, p2), (p2, p3), . . . , (pm−1, pm) without rehashing hubs, starting with kid p1 of the root and consummation with leaf pm in the tree.

2)An Frequent Items Ultrametric -tree is built by embedding's all itemsets as its ways; each itemset contains the identical number of successive things. Along these lines, the greater part of the FIU-tree leaves are indistinguishable tallness.

3)Each child leaf in the FIU-tree is made out of two fields: named thing name and tally. The tally of a thing name is the quantity of exchanges containing the itemset that is the arrangement in a way finishing with the thing name. Non leaf hubs in the FIU-tree contain two fields: named thing name and hub join. A hub connection is a pointer connecting to youngster hubs in the FIU-tree.

## III.SYSTEM REQUIREMENT SPECIFICATION

Frequent itemsets mining (FIM) is the main problem in association rule mining (ARM) and sequence mining. The present mining algorithm for frequent itemset is unavailable for mechanism that facilitate parallelization, load balancing, data distribution and error tolerance on large clusters. Speeding up the process of FIM is critical and indispensable, because FIM consumption accounts for a significant portion of mining time due to its high computation and input/ output (I/O) intensity.

**3.1 User Requirements**

The user requirement for our proposal is big data accessing with MapReduce. The user should be familiar with the concept of data mining, data accessing through FiDoop and Hadoop programming. The user should be familiar with the concept of wireless network, internet and always have the knowledge of uploading and downloading data, the user should have sufficient knowledge memory management of the algorithms used.

**3.2 Functional Requirements**

This describes hoe the user requirements are fulfilled. The proposed technique for data hierarchy mining under MapReduce techniques for Itemsets under search includes paralyzing FIUT (i.e., the frequent items ultra-metric trees method). The algorithm includes modules for establishing handshake between the source and destination.

**3.3 Non-functional requirements**

- Usability: the project will be used in infrastructure wireless network, data analysis and management. This is developed for the dunamic networks under internet.
- Reliability: the architecture is reliable .it takes care that the data sent is received and it includes the authorization and authentication methods to handle the data.
- Maintenance: the operation and development of project for real time applications will be provide to the users.
- Scalability: project is will work well even if the network size is infinite.
- Interoperability: the project works well with respect to continuous sending and receiving of data between remote clients, also from one network to the other.

### 3.6 Specifications

#### 3.6.1 Hardware requirements:

- Processor: Intel family
- Version: Pentium and Higher
- Speed: 2.2GHz
- Memory: 2GB

#### 3.6.2 Software Requirements

- O S: Linux
- Flavor; Ubuntu 10.10 and higher
- Envirnoment: Java JDK 1.6.0_37.20
- Frame work: Hadoop 1.1.1.

### 3.7 System analysis

System analysis part consists of two modules.1.existing system, 2 .proposed system.

#### 3.7.1 Existing System:

The existing system is designed for data mining technique in reserving a given keyword based search for analyzing a given database or file. Each node is aligned with its dedicated tree for parallelism in retrieving.

**Drawbacks:**

1. System under goes execution for series data pattern
2. Modulation and simulative analysis is proposed only for maximum support graph.
3. Support graph is restricted only upto few parametric analysis and hence fail s to retrieve longer value analysis.
4. The system also has restricted limit on loading a file and hence violates raw file as input.

#### 3.7.2 Proposed System:

Data mining has become one of the most challenging tasks in current technological work of analysis and thus we have proposed this methodology of improving the dataset deforming and refiltering the database. This results in load overhead on server, under big data tool using Hadoop cluster, we have proposed this system for achieving an enhanced value of data set for retrieval and thus the same is improved for enhancing system performance.

*Advantages:*

1. The proposed system is designed to fit parameters of existing system.
2. Support minimum and maximum graph is proposed for the system to improve the analyzing efficiency.
3. Confidence support graph is proposed and contributed for increasing the relay performance and analysis
4. Confidence graph provides narrow-down information on fetching data and resource oriented sharing.

Problem Definition:

In a developing nation as India, the economical conditions are below the standard poverty line and hence the medical treatment and making is not possible achieved and hence to achieve the same the proposed system is designed and developed.

## IV.SYSTEM DESIGN

Data Systems Analysis and Design-Development Life Cycle ➔ Businesses and associations use different sorts of data frameworks to strengthen the numerous procedures expected to do their business capacities. Each of these data frameworks has a specific reason or center, and each has its very own existence. This "life of its own" paradigm is known as the frameworks improvement life cycle or System Development Life Cycle(SDLC), and it incorporates the whole

procedure of arranging, building, conveying, utilizing, redesigning, and keeping up a data framework. The improvement of another data framework includes a few diverse, however related exercises. These exercises, or stages, as a rule incorporate arranging, investigation, outline, usage, and upkeep/support. At the end of the day, SDLC is a reasonable model that provides venture administration in data framework improvement.

**4.1 System Architecture**

Fidoop architectural overview is been focused and demonstrated in the following section; fig 4.1 projects the architected diagram.

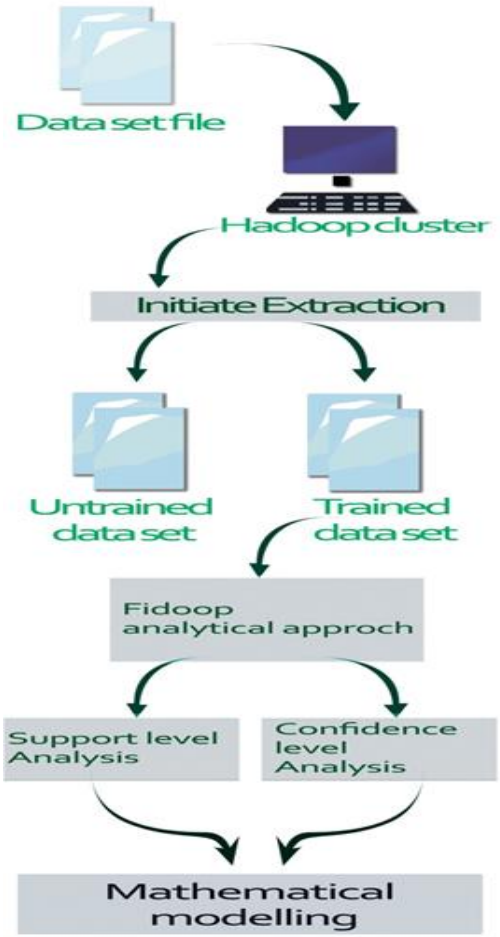


***Fig 4.1: System Architecture***

The system architecture consists of a mining and extraction algorithm for development of system protocol design and analysis. This system is featured to collect the data from the independent sources and project an extraction technique under a privileged authenticated status. The data after extraction is projected under computing state for generation of support and confidence graph. Each graph generated consist of general management and maximum support.

Support and confidence graph projects the overall status on developing and designing the system requirement as per the resource availability. In our proposed system we have discussed about chronic infections and diseases. Each time a stipulated system is generated and thus its acquired results are analyzed and added.

**4.2 METHODOLOGY:**

A. First MapReduce Job

The first MapReduce job is responsible for creating all frequent one-itemsets. A transaction database is partitioned into multiple input files stored by the HDFS over data nodes of a Hadoop cluster. Each mapper sequentially reads each transaction from its local input split, where each transaction is stored in the format of pair<LongWritable offset, Text record>. Then, mappers compute the frequencies of items and generate local one-itemsets.

B. Second MapReduce Job

Given frequent one-itemsets generated by the first MapReduce job, the second subsequent MapReduce job applies a second round of scanning on the database to prune infrequent items from each transaction record. The second job marks an itemset as a k-itemset if it contains k frequent items ($2 \leq k \leq M$, where M is the maximal value of k in the pruned transactions).

C. Third MapReduce Job

The third MapReduce job—a computationally expensive phase—is dedicated to:

1) decomposing itemsets;

2) constructing k-FIU trees;

3) mining frequent itemsets.

The main goal of each mapper is twofold:

1) To decompose each k-itemset obtained by the second MapReduce job into a list of small-sized sets, where the number of each set is anywhere between 2 to $k - 1$ and 2)
2) To construct an FIU-tree by merging local decomposition results with the same length.

**4.3 System Data Flow Diagrams**

The proposed system design of DFD is shown below for overall representation of the Fidoop system under fig, the system allows the data modulation and demodulation approaches I understanding and technically perfuming the mining operations under the given scheme of evaluation. Initially the system is authenticated and then followed by the system behavioral approach of understanding and designing a extraction of datasets. These datasets are medical datasets under active chronic infection. In order to achieve an analytical result, the system has to be developed and designed in maintaining and segregating the system behavioral approach.

On extraction, the system retrieves the datasets from the input stream and the processing is initiated. Under this model of principle the system is aligned to perform the mining operation on big data under chronic infection and syndrome analysis.

The minimal support graph and confidence graph is extracted under the process, the support graph demonstrates the system behavior of analyzing and understanding the system behavioral approach for mining a supportive data under a given category. Such category analysis is performed under confidence graph.

Confidence graph is retrieved from the serial processing of support graph and hence the confidence graph provides the overall beneficiary for system design and development. Under confidence the exact parametric help and assistance is monitored and obsercved for safe and betterment in result prediction.

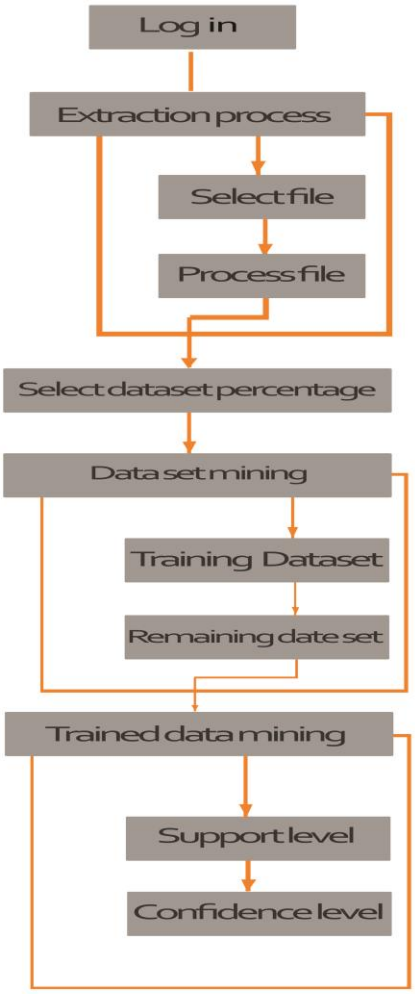


***Fig. 4.3 System data flow dia.***

**4.4 Sequence Diagram**

Sequence diagram, analyses the overall flow of proposed system under scenarios of data mining for analyzing chronic diseases. Each system module is considered fewer than two major attributes as FiDoop Processing and Hadoop Analysis, future Analysis is split into Technique Processing and development.

The below mentioned sequence diagram is programmed to achieve an overall system behavioral cum analysis approach for masking and demasking the dataset, under this the mathematical model is recalled. The system of hyperactive analysis is performed. The datasets are collected and extracted in the initial sets and thus the main contribution is retrieved under the mathematical approach.

The mining approach is appended on the medical datasets for acquiring and analyzing the data segment under chronic infection with major attributes. Each attribute is subjected to the overall behavior of the system. But as the time intently is computed, the system is reanalyzed and mining based support and confidence graph is achieved.

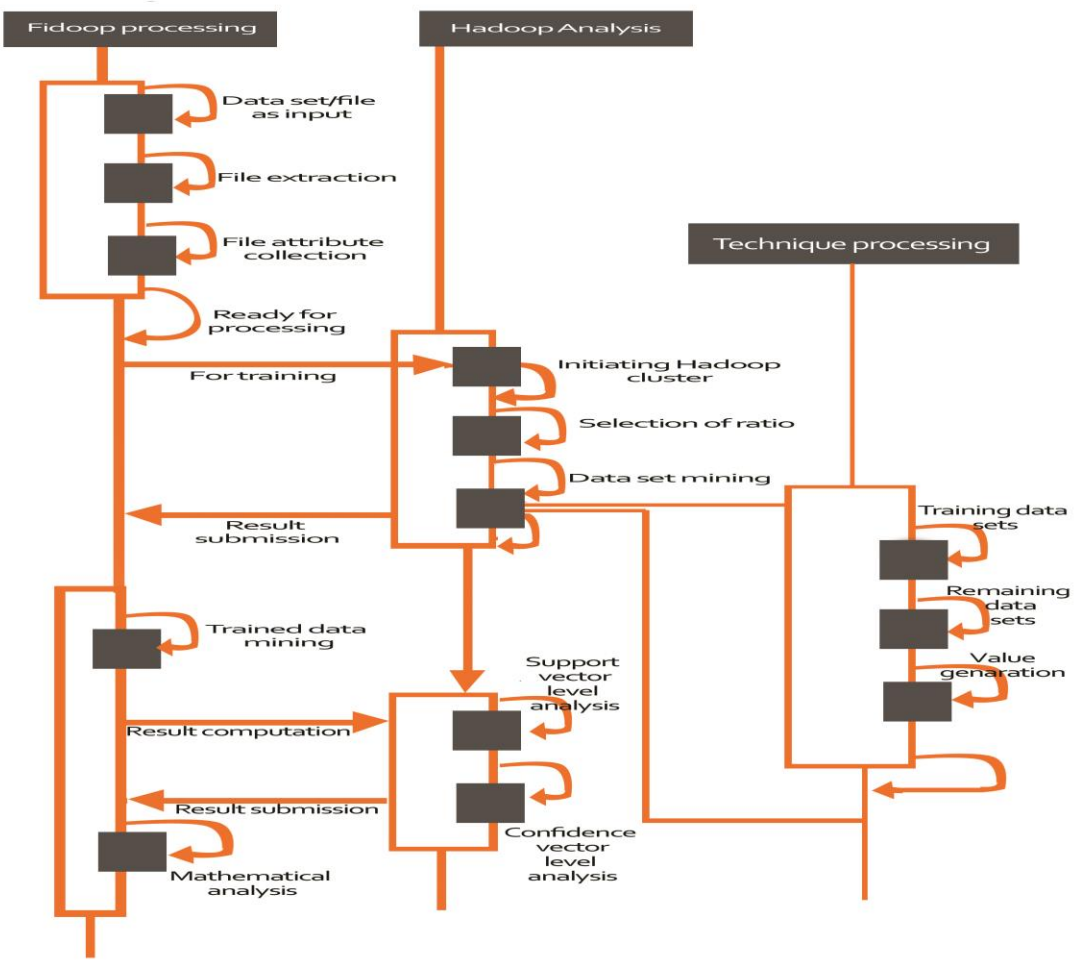


***Fig 4.4: Sequence Diagram***

## V.IMPLEMENTATION

The process of converting a standard system prototype into a practically working module is demonstrated in this section, basically a Hadoop cluster is installed under Ubuntu environment. For the process of implementation and analyzing the efficiency of the system, we have discussed the input dataset as the chronic infection sets under active and structured format.

**Step 1: Data set Initialization**

***f*** :file to upload

***K*** : itemsets generation during the process of mining

***n :*** all itemsets generated during the process of mining

***T*** : tree (FIU) infrastructural tree with dynamic slotting

***U*** : entire file for data mining from a data base

***A :***Selected Percentage of file for mining under a synchronized manner

***B*** :Unselected Percentage of file for mining under a synchronized manner

The above taken variable are computed under the data allocation scheme of understanding and parametric analysis, the mathematical model is algorithmically explained

**Step 2: FIUT Tree Generation and Process initiation**

FIUT($\prod$) Tree Generation

T={K}considering an instance of T for generation of K item sets under the given input sample file for each of K is as shown below.

where, K={ $K_1, K_2, K_3, K_4, \ldots\ldots K_n$ }

Tree generation process

R : root node / R $\in$ T && R $\in$ {K}

R→$R_1, R_2$ / $R_1, R_2 \subseteq$ R && $R_1, R_2$ !=R

$R_1, R_2 \subseteq$ M /M={$M_1, M_2, M_3, M_4 \ldots\ldots M_n$}

**(M,R)=K**

**Final accessing of Load Balancing**

Therefore, $$(M,R) = K\sum_{i=0}^{n} \frac{(\mathrm{Ri})}{(Mi - Ri)}$$ ----------**eq. (1)**

$$\mathrm{T}=\int_0^n (\mathrm{K}):(\mathrm{Ri})/(\mathrm{Mi}\text{-}\mathrm{Ri})$$

----------**eq. (2)**

**Step 3: Itemset Generation**

On considering a dataset under database module of an desired data mining server under Hadoop single node cluster as

$DB_i$= DataBase

$\Pi$ = Transaction under DataBase

$T_f \subseteq DB_i$ / $T_f$< 20% ( $DB_i$)

For all

$\Pi$ → for i=0 to k

{

For each of 'k'→ $\prod$

$\prod_i$ : fetch($K_i$)

call step 2 : FIUT($\prod$)

**Step 4: Analysis**

Under the generated tree, a flatter analysis of groped datasets is considered and mined as

FIUT($\prod$)--------**eq (3)**

$$\{\int_{0}^{n} (K):( Ri)/(Mi-Ri)$$

Compute confidence and support graph.

Ration Selection ( $R_i$) [0~1]

{

Generate,

$$\{\int_{0}^{n} (K)\pi$$

------eq. (4)

Show graph (display); }

## VI.TESTING

The fundamental point of framework testing is to decide blunders amid execution of the venture. The information are the inputs which are utilized to test the framework. Test cases are utilized to watch the execution of the framework. Experiment guarantees that the framework is watched for all conceivable inputs. The normal execution of the framework can be considered on various mixes. In this way test cases are chosen which have inputs and the yields are on expected lines, inputs that are not legitimate and for which appropriate messages must be given and inputs that don't happen oftentimes which can be viewed as exceptional cases.

### 6.1 Unit Test

Test cases are built in order to test and make sure that all the components within the system interact properly. Unit testing focuses on testing the each module in the system. The goal is to detect the error in each module. The various test cases for the modules of the project are listed as below.

- **Items to be tested** – In case of unit testing the items is Registration Module, Login page, uploading file, Key generation when user request forfile, Downloading file, Admin authentication, updating a file.
- **Successful/unsuccessful Criteria** – The pass or fail criteria are designed to check whether it is working properly or not.

#### 6.1.1.1 Unit test for Registration module

The test case for registration module is given below table 6.1 which tells the name of the test, inputs, expected output, actual output remarks. If this test gives expected output then the system performance is good. Admin registration contains all the information about user. User information is stored in the database.

**Table 6.1: Unit test case for Registration module**

| | |
|---|---|
| **Test case** | 1 |
| **Test Name** | Unit test for Registration |

| | |
|---|---|
| **Item being tested** | Registration page |
| **Input** | Name<br>Password |
| **Expected Output** | Registration should be successful |
| **Obtained Output** | The obtained output is same as expected output as shown in snapshot A1 |
| **Remark** | Successful |

The table 6.1 shows the test case for user registration. The user registration contains varies fields like name, password. User needs to fill the required fields then user should click on register button if all the fields correct then it will show registration/login is successful otherwise it will display registration unsuccessful. It store user information in database. If the user name and email address already exist in database then it will display user/email already exists. Otherwise registration is successful. The different fields like name password, email, mobile number should match the pattern otherwise it will display invalid name and email. If all fields are filled correctly then it registration is completed. Here, the obtained output is registration is successful and expected output is also same as shown in snapshot 6.1.

**6.1.1.2 Unit test for empty registration form**

The table 5.2 shows the test case for user registration form. If the user does not enter the required field then it will display the message like empty fields are not allowed. If the user enter the wrong email address it will display then enter valid email address. If the user name is wrong then it will display enter valid user name. If the mobile number is wrong then it will display enter valid mobile number.

**Table 6.2: unit test for empty registration form**

| | |
|---|---|
| **Test case** | 2 |
| **Name of Test** | Unit test of registration form |
| **Item being tested** | Registration page |
| **Input** | No inputs |
| **Expected Output** | Registration successful |
| **Obtained Output** | Empty fields are not allowed is the obtained output is as shown in snapshot 6.2 |
| **Remark** | Unsuccessful |

If the user does not give any input in the registration form then it will display empty fields are not allowed. Here, the obtained output is not same as the expected output because expected output is registration should be successful but obtained output is empty fields are not allowed as shown in snapshot 6.2.

**6.1.1.3Unit test for Main page**

The table 6.3 shows the unit test for main page. User should provide the file to upload in this section.

**Table 6.3: Unit test for Main page**

| | |
|---|---|
| **Test case** | 3 |
| **Name of Test** | Unit test of Main Page |
| **Item being tested** | Main Page |
| **Input** | Search File from Cluster |
| **Expected Output** | Accept file and move towards mining |
| **Obtained Output** | Obtained output is successful and goes to file list page. it is same as expected output as shown in snapshot 7.4 |
| **Remark** | Successful |

## VII.SNAPSHOTS

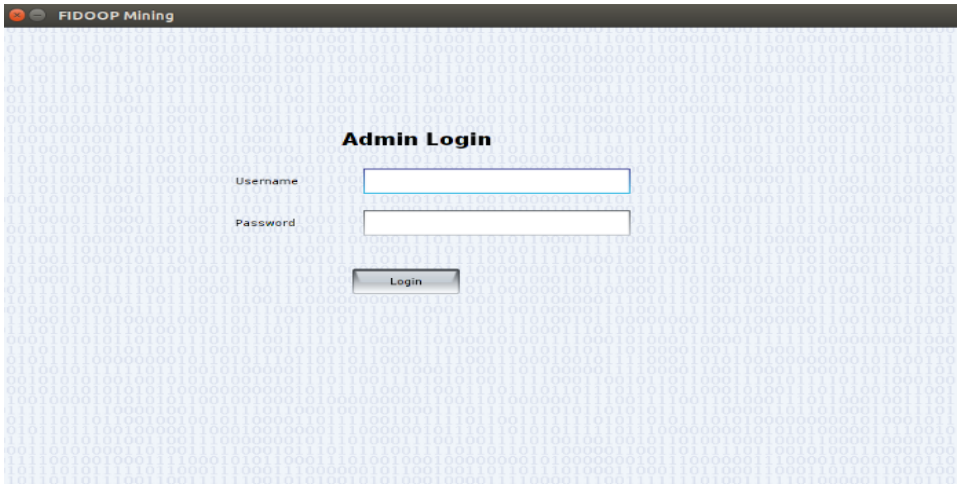


***Fig S1 Admin Login***

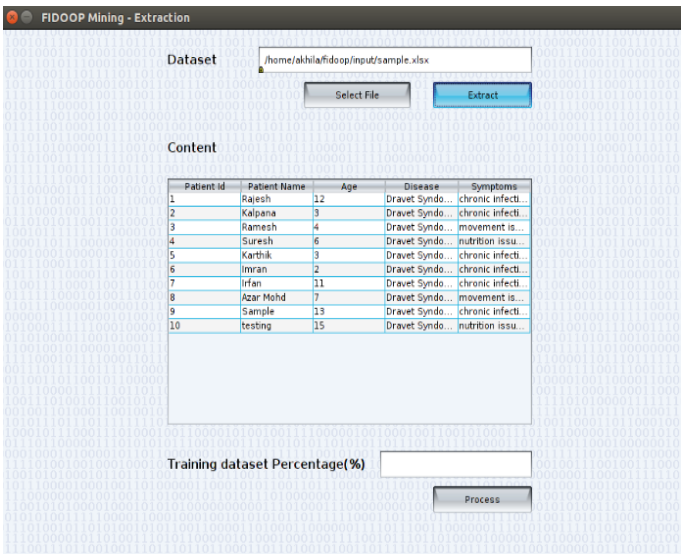

*Fig S2 Selection of input files from dataset & Extract*

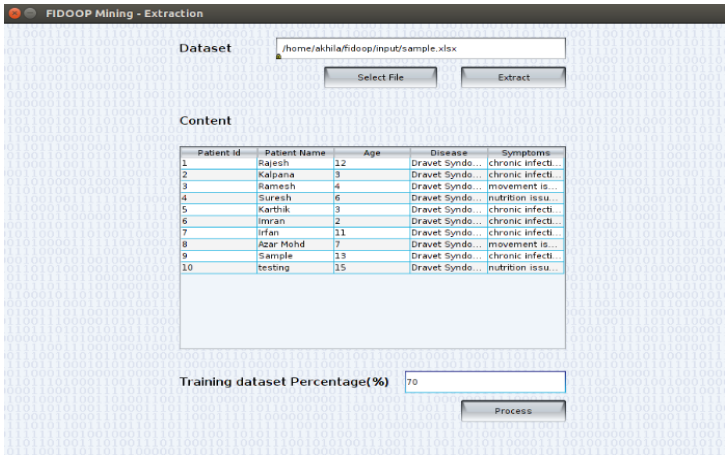


*Fig S3 Selection of training dataset percentage*

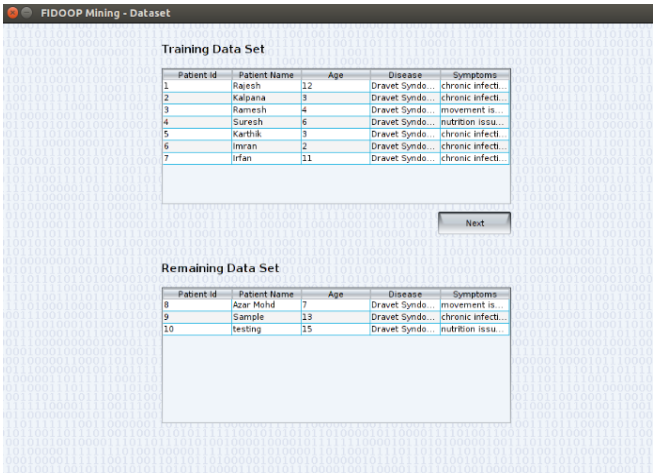


*Fig S4 Result of extraction-Training data set and Remaining data set*

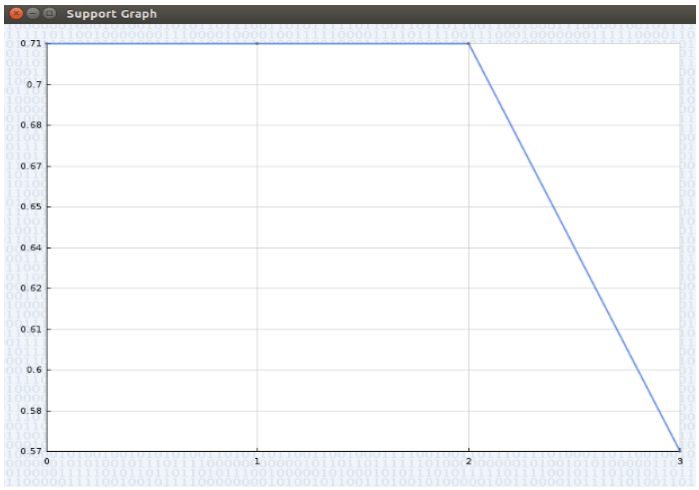


*Fig S5 support graph*

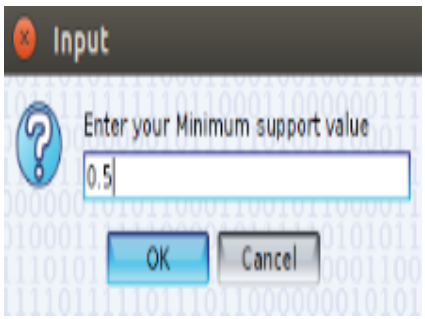


*Fig S6 minimum support graph*

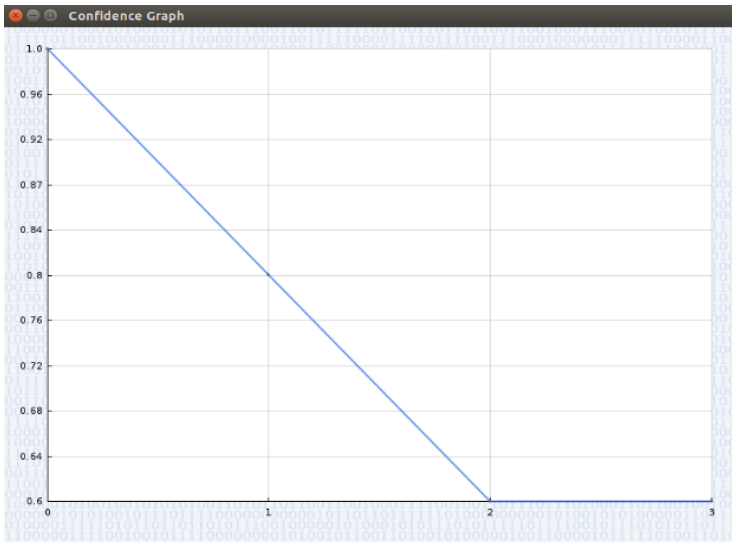


*Fig S7 confidence graph*

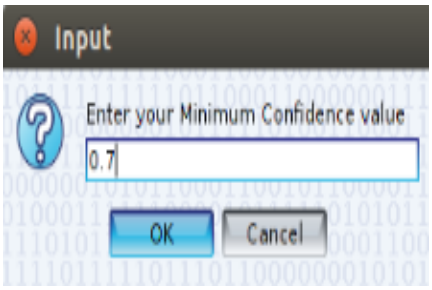


*Fig S8 minimum confidence graph*

## VIII. CONCLUSION AND FUTURE ENHANCEMENT

Fidoop system has been dedicated to produce an accurate data mining results under Hadoop single node cluster environment, the system is simulated under Ubuntu for easy and high expert assistance. The proposed system under implementation shall produce appropriate results of support and confidence graph, the system support graph represents the high scale availability of the system under a random operating range and high confidence is been projected under confidence graph.

The system achieves high efficiency gain for providing static information resources for dynamic and critical data under big data mining. Results are detailed and discussed in previous chapters with overall system design and analysis. This system in future can be enhanced with a diplomatic sentiment anslysis and redefine process of computation under big data environment. Apparently the proposed project can be used and appended in medical static data analysis and also the medical crisis and resource sharing analysis. This application is highly simulative and is active on all the medical conditions and diseases v/s resource mapping and decision analysis can be fetched.

## ACKNOWLEDGEMENT

I am highly obliged to Department of computer science and engineering, UBDT college of engineering. And I am highly grateful and thankful to our guide Prof .B.N.Veerappa for his valuable instructions, guidance, corrections in my project work and presentation.

## REFERANCES

[1] M. J. Zaki, “Parallel and distributed association mining: A survey,” *IEEE Concurrency*, vol. 7, no. 4, pp. 14–25, Oct./Dec. 1999.

[2] I. Pramudiono and M. Kitsuregawa, “FP-tax: Tree structure based generalized association rule mining,” in *Proc. 9th ACM SIGMOD Workshop Res. Issue Data Min. Knowl. Disc.*, Paris, France, 2004, pp. 60–63
.

[3] R. Agrawal, T. Imieli´nski, and A. Swami, "Mining association rules between sets of items in large databases," *ACM SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, 1993

.

[4 ] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Min. Knowl. Disc.*, vol. 8, no. 1, pp. 53–87, 2004.