

# **MITIGATING THE RISK OF CUSTOMER CHURN AND MANAGEMENT RECOMMENDATION**

*A project report submitted in the partial fulfillment of the requirement for the award of the degree  
of*

## **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

**By**

**A.S.Amrutha**  
**(1210314201)**

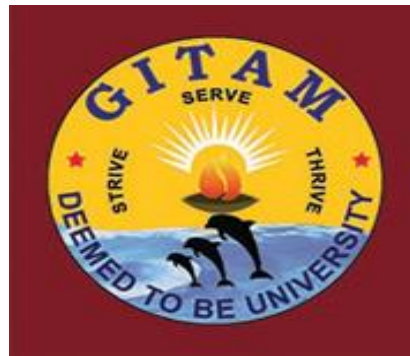
**J.Anusha**  
**(1210314202)**

**Kausar**  
**(1210314217)**

Under the Esteemed Guidance of

**Smt.G.L.Aruna Kumari**

**Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM(DEEMED TO BE UNIVERSITY)**

**(Estd. u/s 3 of the UGC Act, 1956)**

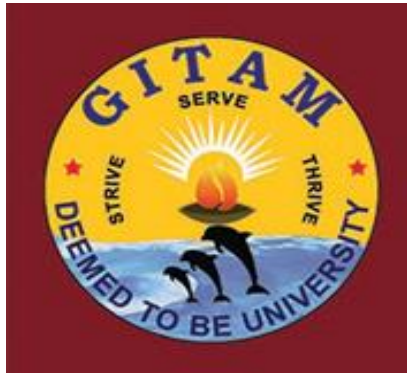
**VISAKHAPATNAM-530045**

**ANDHRA PRADESH**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**GITAM Institute Of Technology**

**Visakhapatnam-530045**



**CERTIFICATE**

This is to certify that the project report entitled “**MITIGATING THE RISK OF CUSTOMER CHURN AND MANAGEMENT RECOMMENDATION**” is the bonafide work carried out by **A.S.Amrutha(1210314201)**, **J.Anusha(1210314202)**, **Kausar(1210314217)** in partial fulfillment of the requirement for the award of the degree of Bachelor Of Technology in Computer Science and Engineering, GITAM Institute of Technology, GITAM Deemed to be University during the academic year 2017-2018. This project work is original and was not submitted earlier for the award of any degree of any institution.

**HEAD OF THE DEPARTMENT**

**Dr.KONALA THAMMI REDDY**

Professor,Dept. of C.S.E  
GITAM UNIVERSITY

**PROJECT GUIDE**

**Smt G.L.ARUNA KUMARI**

Assistant Professor  
Department of CSE

## **DECLARATION**

We (**A.S.Amrutha, J.Anusha, Kausar**) hereby declare that this project entitled” **Mitigating The Risk Of Customer Churn And Management Recommendation**” is an original and authentic work done in the Department of Computer Science and engineering, GITAM Institute of Technology, GITAM Deemed to be University, submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science Engineering.

<b>ROLL NO.</b>	<b>NAME</b>	<b>SIGNATURE</b>
1210314201	A.S.Amrutha	
1210314202	J.Anusha	
1210314217	Kausar	

## **ACKNOWLEDGEMENT**

We express a profound sense of gratitude to our project guide **Smt G.L.Aruna Kumari**, Assistant Professor, Department of Computer Science and Engineering, GITAM institute of Technology, GITAM Deemed to be University for the expert guidance and motivation given to us throughout our work. She spared her valuable time patiently listening to our problems and helping us in finding solutions. The project report would not have been shaped to this form without her constant encouragement and co-operation.

We also express my sincere thanks to **Dr. Konala Thammi Reddy**, Head of the Department, Department of Computer Science and Engineering, GITAM Deemed to be University for providing a great support for us in completing our project by arranging the trainees, faculty needed to complete our project and for giving me the opportunity of doing the project.

We would like to express our sincere thanks to all the faculty members, lab programmers of Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM Deemed to be University for helping us to complete our project.

We also express our sincere thanks to **Prof. K. Lakshmi Prasad**, Principal, Institute of Technology, GITAM Deemed to be University for his help given to us during the project.

We would like to thank our parents for inspiring us all the way for arranging all the facilities and resources needed for our project. Not to forget, my non-teaching staff, my friends who had directly and indirectly helped and supported me in completing my project in time.

## **ABSTRACT**

As market competition intensifies, customer churn management is increasingly becoming an important means of competitive advantage for companies. However, when dealing with big data in the industry, existing churn prediction models cannot work well. In addition, decision makers are always faced with imprecise operations management. In response to these difficulties, a new clustering method(SDSCM) is proposed. Experimental results indicate the SDSCM has stronger clustering semantic strength than subtractive clustering method(SCM) and fuzzy c-means(FCM). Then, a proposal SDSCM algorithm is implemented through a Java framework. In the case study, the proposed parallel SDSCM algorithm enjoys a fast running speed when compared with the other methods. Furthermore, we provide some marketing strategies in accordance with the clustering results and a simplified marketing activity is simulated to ensure profit maximization.

### **PROJECT MEMBERS**

A.S.AMRUTHA

(1210314201)

J.ANUSHA

(1210314202)

KAUSAR

(1210314217)

### **PROJECT GUIDE**

(G.L.ARUNA KUMARI)

# TABLE OF CONTENTS

<b>1. INTRODUCTION.....</b>	<b>08</b>
<b>2. LITERATURE REVIEW .....</b>	<b>13</b>
2.1: Problem Analysis .....	13
2.2: Axiomatic Fuzzy Set.....	14
2.2.1: Calculation .....	16
2.3: Subtractive Clustering Method .....	17
2.3.1: K-Means .....	28
2.4: Semantic Driven Subtractive Clustering Method .....	19
2.4.1: Algorithm Of SDSCM .....	20
2.4.2: Time Complexity Analysis.....	23
2.5: Implementation Of SDSCM And K-Means With Map Reduce.....	24
2.5.1: Deploying SDSCM With Map Reduce .....	24
2.5.2: Algorithm Of Parallel SDSCM.....	25
2.5.3: Time Complexity Analysis.....	27
2.6: Flowcharts .....	28
2.6.1: Flowchart Of Clustering Methods .....	28
2.6.2: Churn Management Framework .....	29
<b>3. SYSTEM ANALYSIS.....</b>	<b>30</b>
3.1: Software requirements Specifications .....	30
3.1.1: Functional Requirements .....	30
3.1.2: Non-Functional Requirements .....	30
3.2: Feasibility Study .....	30
3.2.1: Economic Feasibility .....	31
3.2.2: Technical Feasibility .....	31

3.3: Exisisting And Proposed System.....	31
3.4: System Configuration .....	32
<b>4. SYSTEM DESIGN.....</b>	<b>33</b>
4.1: Use Case Diagram.....	33
4.2: Sequence Diagram .....	34
4.3: Activity Diagram .....	35
4.4: Class Diagram.....	36
4.5: Collaboration Diagram .....	37
<b>5. SYSTEM IMPLEMENTATION.....</b>	<b>38</b>
5.1: LoadData.java .....	38
5.2: Cluster.java .....	40
5.3: KmeanCluster.java.....	40
5.4: KmedoidCluster.java .....	43
5.5: Sdscmcluster.java .....	44
<b>6. TESTING.....</b>	<b>47</b>
<b>7. RUNNING THE SYSTEM.....</b>	<b>50</b>
7.1 ScreenShots .....	50
<b>8. CONCLUSION.....</b>	<b>61</b>
<b>9. REFERENCES.....</b>	<b>62</b>

# 1. INTRODUCTION

Nowadays, expanding market shares has become more and more tough for service industry, such as telecommunications industry, as the competition is fierce and market is increasingly saturated. Thus, these companies pay more attention to the existing customers so as to avoid customer churn. Customer churn refers to the loss of customers who switch from one company to another competitor within a given period. Industrial practice has shown that customer churn can lead to huge economic losses and even hurt the company's public image. Hence, customer churn management is extremely important especially for the service industry.

There exist many effective ways in the literature for handling customer churn management problem. Analytical methods mainly include statistical models, machine learning, and data mining. Castro and Tsuzuki propose a frequency analysis approach based on k-nearest neighbors machine learning algorithm for feature representation from login records for churn prediction modeling. Au et al propose a new datamining algorithm, called data mining by evolutionary learning (DMEL), to handle classification problems. Moreover, it is applied to predict churn under different churn rates with telecom subscriber data. Decision tree, neural network, and k-means are selected by as main techniques to build predictive models for telecom customer churn prediction. Their empirical evaluation indicates that data mining techniques can effectively assist telecom service providers to make more accurate customer churn prediction. Verbraken et al. formalize a cost-benefit analysis framework and define a new expected maximum profit criterion. This general framework is then applied to the customer churn problem with its particular cost-benefit structure. Recently, based on a boosting algorithm, a robust churn prediction model has been successfully applied in churn prediction in the banking industry. Although these methods can deal with customer churn problem efficiently, we should also notice that they are limited to process small structured data, such



as account data and call details data, all of which are less than ten thousand records. However, with the widespread adoption of smart phones and growth in mobile internet, companies today have accumulated unprecedented amounts of data sources. Take China Telecom as an example, as of July 2015, the system generates 10.5 trillion user-domain data records, and the corresponding data storage amount is hundreds of terabytes (TB) per day. The huge volume of data has the typical features of big data , and is hence referred to as “telco big data,” which are the data to be analyzed in this paper and include call detailed records, Internet traffic logs, user profiles, location updates, social networking information so on and so forth. On account of big data have 55% possibility to bring the most value to operators in the area of customer retention, companies are eagerly seeking big data analytics solutions to solve customer churn problem for the sake of turning the data into valuable business insights.

In fact, the use of industrial big data for customer churn management has caught researchers’ eyes because traditional methods are not engineered for the type of big, dynamic, and unstructured data. For instance, Cloudera introduces some big data use cases for telcos including customer experience management, network optimization, operational analytics, and data monetization. Huang et al. empirically demonstrate that telco big data make churn prediction much easier through 3V’s perspectives. Actually, the difficulty for customer churn management lies in selecting analytical algorithms which allow them to cope with industrial big data. However, there exist a few literature studying big data clustering algorithms. Thus, with the purpose of developing the efficient algorithm in view of customer churn management, we first notice that there are two major challenges when utilizing telco big data.

- 1) It is difficult to give an explicit definition for the fuzzy concept “customer churn.” For example, how to define customers of telecom operators have lost? Is it elimination of telephone numbers? In arrears for more than 3 months (postpaid customers)? Or no calling within 3

months (prepaid customers)? Moreover, with the telco big data, identifying the characteristics of churning customers is more challenging because a customer's decision of whether to churn is also affected by the social networking information, such as comments of friends and recommendations of a celebrity. Therefore, how to efficiently mine those unstructured social networking data, and thus identify the churning customers? That is, how to define the "customer churn" on the basis of social networking information?

2) The existing analytical methods do not work very well in dealing with big data. First, as mentioned earlier, in order to fully define "customer churn," we need to efficiently mine the unstructured social networking data. Because the relational database which stores traditional data effectively cannot process the unstructured data, so do the analytical methods. Therefore, we need to propose a new distributed computing method which is able to process the data stored in a specific infrastructure. Second, the volume of telco big data has reached a TB level. For example, in telco big data, business support systems (BSS) and operations support systems (OSS) sources are around 2.3 TB new coming data per day, such as real time billing and call details' information, unstructured information like textual complaints, mobile search queries, and trajectories. Thus, identifying the churning customers is extremely difficult. As the traditional analytical methods are more likely to encounter performance bottlenecks when conducting customer churn analysis, the new analytical method requires to be as accurate as possible. Therefore, to maximize the value of telco big data, proposing an efficient and appropriate parallel algorithm is a big challenge.

To solve the first problem, we introduce axiomatic fuzzy sets (AFSs) to generalize the definition of customer churn. The key idea of AFS method is that several simple concepts or attributes can express many complex concepts by AFS algebra and AFS structure [12]. Moreover, it can deal with the preferences' attributes, which reflects the preference extent of a customer influenced by social networking. For

instance, let  $X = \{x_1, x_2, x_3, x_4, x_5\}$  be a set of five customers.  $M = \{m_1, m_2, m_3, m_4, m_5\}$  is a set of attributes, where  $m_1$  = elimination of telephone numbers,  $m_2$  = without phone calls in 3 months,  $m_3$  = without data traffic in 3 months, and  $m_4$  = in arrears for more than 3 months. For a fuzzy concept  $\eta = m_1 + \{m_2, m_3\} + m_4$ , the semantic significance of  $\eta$  is “persons who have eliminated telephone numbers,” “persons without phone calls and data traffic in 3 months,” or “persons in arrears for more than 3 months,” which represents “customers who are more prone to churn.”

As for the second challenge, motivated by subtractive clustering method (SCM) and AFS, we propose a new big data clustering algorithm called semantic-driven SCM (SDSCM). Although the distributed k-means algorithm is popular, the clustering results are likely to be imprecise if the initial parameters are valued improperly. In comparison, SCM is usually used to generate more precise input parameters for k-means based on raw data, including cluster centroids and clustering number. Nevertheless, there exist many uncertain parameters in SCM, which leads to the clustering inaccuracy. Hence, to improve its accuracy, the parameters of SCM are determined automatically in SDSCM. Furthermore, a parallel SDSCM algorithm is implemented through a Java MapReduce framework for real-time and efficient data analysis.

The three main contributions of this paper are as follows. First, we propose a new algorithm called SDSCM, which improves clustering accuracy of SCM and k-means. Moreover, this algorithm decreases the risk of imprecise operations management using AFS. Second, to deal with industrial big data, we propose a parallel SDSCM algorithm through a Java MapReduce framework. Third, in the case study of China Telecom, the results show that the parallel SDSCM and parallel k-means have high performance, when compared with traditional methods.

This paper is organized as follows. In Section II, urgent problems faced by companies are described and some basic theories are

introduced. In Section III, a new clustering algorithm called SDSCM is proposed. In Section IV, some evaluation indexes are introduced and experiments are conducted on standard data sets to compare the performance of SDSCM, SCM, and fuzzy C-means (FCM). In Section V, modifications of SDSCM and k-means with MapReduce are presented. In Section VI, the distributed clustering methods are implemented to address the problem of customer churn faced by China Telecom. The last Section is for conclusion and future research.

## **2. LITERATURE REVIEW**

### **2.1 PROBLEM ANALYSIS:**

Our project provides new methods to help the company better mitigate the risk of customer churn and hence to gain higher profits. It mainly studies customer churn problem in service industry under the big data environment. Specific problems include as follows.

- 1) How to define the fuzzy concept “customer churn” comprehensively?
- 2) Since many parameters affect the accuracy of SCM, how to obtain their values properly based on raw data?
- 3) How to evaluate the semantic strength (SS) of algorithms? How to design a new clustering algorithm by integrating semantic fuzzy concept with SCM, which performs better than FCM and SCM?
- 4) To deal with big data sets effectively, how can we modify the new algorithm be to a parallel one?
- 5) How to use the clustering results to guide the customer churn management of the company? And which customers cluster is the marketing target?

To tackle these problems, we first need some basic theories.

## 2.2 AXIOMATIC FUZZY SET:

AFS is an effective way to describe the fuzzy concept. The membership functions and their logic operations are determined by original data and facts instead of intuition. Moreover, several simple concepts or attributes can express many complex concepts using AFS algebra and AFS structure. The AFS structure and the membership function are defined as follows.

Let  $(M, \tau, X)$  be an AFS structure, where  $X$  is the universe of discourse, and  $M$  is a set of attributes or concepts on

$$X. \tau(x, y) = \{m | m \in M, (x, y) \in R_m\} \in 2M,$$

where  $R_m$  is a preference relation. For  $A \subseteq M$  and  $x, y \in X$ , the following symbol is defined:

$$A\tau(x) = \{y | y \in X, \tau(x, y) \supseteq A\} \dots \dots \dots (1)$$

Let  $X$  and  $M$  be sets,  $(M, \tau, X)$  be an AFS structure and  $(M, \sigma, m)$  be a measure space, where  $m$  is a finite and positive measure with  $m(X) > 0$ . Then,  $(M, \tau, X, \sigma, m)$  is called as a sim-cognitive field. If  $A_k(\{x\}) \in \sigma, \forall k = 1, 2, \dots, n, x \in X$ ,

$k=1$   $A_k$  is a measurable concept of  $(M, \tau, X)$  under  $\sigma$  whose membership function is defined as follows:

$$\mu_{\bigcup_{k=1}^n A_k}(x) = \sup_{1 \leq i \leq n} (m(A_i(\{x\}))/m(X)) \quad \forall x \in X \dots \dots (2)$$

Membership  $\mu$  represents the degree that  $x$  belongs to the concept, which means the higher  $\mu$  is, the closer  $x$  is to the concept.

Assume  $m(A) = |A|$  ( $|A|$  is the number of elements in set  $A$ ). The fuzzy set membership function of (2) is completely determined by subpreference relations of the simple concepts in  $M$ .

For example, as shown in Table I, let  $X = \{x_1, x_2, x_3, x_4, x_5\}$  be a set of five customers.  $M = \{m_1, m_2, m_3, m_4, m_5, m_6\}$  is a set of attributes, where  $m_1$  = peak calls,  $m_2$  = off - peak calls,  $m_3$  = weekend calls,  $m_4$  = national calls,  $m_5$  = international calls, and  $m_6$

= loyalty. The subpreference relation of  $m_6$  is expressed as  $x_1 > x_2 = x_5 > x_4 > x_3$ , where  $x_1 > x_2$  means  $x_1$  is more loyal than  $x_2$ . For or a fuzzy concept  $\eta = \{m_1, m_2, m_3\} + \{m_4, m_5\} + m_6$ , the semantic significance of  $\eta$  is “persons with many calls during peak time and off-peak time and weekend,” “persons with many national and international calls,” or “high loyal persons,” which represents “customers who are less prone to churn.”

TABLE I  
DESCRIPTIONS OF ATTRIBUTES

	PEAK CALLS	OFF-PEAK CALLS	WEEKEND CALLS	NATIONAL CALLS	INTERNATIONAL CALLS
$X_1$	73	31	1	105	0
$X_2$	54	9	14	77	2
$X_3$	57	32	6	95	1
$X_4$	25	21	1	47	9
$X_5$	57	6	20	83	0

### 2.2.1 CALCULATION:

According to (2),  $\eta(x_i)$  and the membership  $\mu\eta(x_i)$  of each customer can be calculated as follows:

$$\eta(x_1) = \{x_4\} \{m_1, m_2, m_3\} + \{x_5\} \{m_4, m_5\} + \{x_2, x_3, x_4, x_5\} \{m_6\}$$

$$\eta(x_2) = \emptyset \{m_1, m_2, m_3\} + \emptyset \{m_4, m_5\} + \{x_3, x_4, x_5\} \{m_6\}$$

$$\eta(x_3) = \{x_4\} \{m_1, m_2, m_3\} + \{x_5\} \{m_4, m_5\} + \emptyset \{m_6\}$$

$$\eta(x_4) = \emptyset \{m_1, m_2, m_3\} + \emptyset \{m_4, m_5\} + \{x_3\} \{m_6\}$$

$$\eta(x_5) = \emptyset \{m_1, m_2, m_3\} + \emptyset \{m_4, m_5\} + \{x_3, x_4\} \{m_6\}$$

$$\mu\eta(x_1) = m(x_2, x_3, x_4, x_5)/m(x_1, x_2, x_3, x_4, x_5)=0.8$$

$$\mu\eta(x_2) = m(x_3, x_4, x_5)/m(x_1, x_2, x_3, x_4, x_5)=0.6$$

$$\mu\eta(x_3) = m(x_4, x_5)/m(x_1, x_2, x_3, x_4, x_5)=0.4$$

$$\mu\eta(x_4) = m(x_3)/m(x_1, x_2, x_3, x_4, x_5)=0.2$$

$$\mu\eta(x_5) = m(x_3, x_4)/m(x_1, x_2, x_3, x_4, x_5)=0.4.$$

It is obvious that  $x_1$  has the largest membership value, which means  $x_1$  is closest to the semantic concept. That is,  $x_1$  is loyal and less prone to churn. On the contrary,  $x_4$  has the smallest membership value, which means  $x_4$  is prone to change brands and has a high probability to churn.



## 2.3 SUBTRACTIVE CLUSTERING METHOD:

In this section, we introduce SCM for computing the cluster centroids, which belongs to unsupervised learning and can quickly determine the number of clusters and cluster centroids based on the raw data. In SCM, each data point is considered as a potential cluster centroid and the potential is computed as

$$Ml(x_i) = \frac{n \exp -\|x_i - x_j\|^2}{(\tau_1/2)^2} \dots\dots\dots (3)$$

$J=1$

where  $\tau_1$  is the neighbor radius, which influences the scope of a cluster centroid. The larger the  $\tau_1$  is, the greater its impact will be. Thus, the data point with maximum mountain function is the first centroid. Then update the mountain function of each data according to the following equation:

$$Ml(x_i) = Ml(x_i) - M_{i-1}^* \exp -\|x_i - x_1^*\|^2 / (\tau_2/2)^2 \dots\dots\dots (4)$$

where  $\tau_2$  is the influencing weight of the last cluster centroid. Data points near the first cluster centroid will have greatly reduced potential and thus unlikely to be the next cluster centroid. To avoid getting close cluster centroids, according to in general,  $\tau_2 = 1.5\tau_1$ .

### 2.3.1 K-MEANS:

k-means is the simplest one among all these clustering methods. Hence, we introduce it here for computing the clusters. Note that the clustering results are likely to be imprecise if having improperly valued initial parameters in k-means. On the contrary, SCM can generate more precise input parameters based on raw data, including cluster centroids and clustering number [13], [19]. Consequently, in this paper, the parameters generated by SCM pass to k-means so as to improve the accuracy of k-means.

The algorithm of k-means starts with initialized k cluster centroids. Then, data are iteratively assigned to the nearest cluster and the new centroids of k clusters are recalculated until the termination conditions are reached.

## 2.4 SEMANTIC DRIVEN SUBTRACTIVE CLUSTERING METHOD:

For solving the first two problems in Section II, we integrate AFS and SCM, which formed the new algorithm, SDSCM. The innovation points of SDSCM are as follows.

- 1) It can fully express semantic signification of fuzzy concept.
- 2) It can automatically determine the neighbor radius and the weight coefficient of SCM.
- 3) It sets a termination condition on the basis of an earlier study reasonably.

The procedure of SDSCM is shown in Flow chart. First, we use AFS to select related attributes for expressing the fuzzy concept by its membership function and logic operations. Second, according to the calculated membership, we determine the neighbor radius and the weight coefficient of SCM automatically. Third, we use SCM to compute the cluster number and centroids by selecting and updating mountain functions. In this paper, we integrate SCM and AFS as SDSCM. Finally, we use k-means to calculate the clusters with the cluster centroids obtained by SDSCM. The details of SDSCM algorithm are shown as follows.

#### 2.4.1 ALGORITHM OF SDSCM:

The below table shows the description of notation used by SDSCM.

**TABLE  
NOTATION USED BY SDSCM**

SYMBOL	DESCRIPTION
$\mathcal{D}$	Fuzzy concept
$x_i$	$x_i \in X$ and $x_i^F$ is the first centroid in paramete determination. $x_1^*$ is the first centroid in SCM
$\mu_n(x_i)$	Membership of $x_i$
$p_i$	Sum of absolute differences of $\mu_n(x_i)$ and $p_i^F$ is the minimum
$d_i$	Euclidean distance between the first cluster centroid and the other data points. $\bar{d}$ is the mean.
$\tau_1$	Neighbour radius
$\tau_2$	Weight coefficient
$L$	Cycle index
$M_i$	Mountain function and $M_l^*$ is the maximum in l cycle
$E$	Termination condition

**Algorithm 1:** the algorithm of SDSCM

Step 1: According to the fuzzy concept  $\eta$  given by the user, use (5) to compute the membership of  $x_i$

$$\mu_{\eta}(x) = \text{SUP}(m(\eta)/m(X)) \quad \forall x \in X \dots\dots\dots (5)$$

Step 2: Compute the sum of absolute differences of  $\mu_{\eta}(x_i)$   
As

$$P_i = \sum_k^n |\mu_{\eta}(x_i) - \mu_{\eta}(x_k)| \dots\dots\dots (6)$$

Step 3: Select the minimum value of  $p_i$  as the first cluster centroid

$$P_i^F = \min\{p_i\} \dots\dots\dots (7)$$

$$X_i^F = x_i \dots\dots\dots (8)$$

Step 4: Compute the Euclidean distance between the first cluster centroid and other data points. The neighbor radius  $\tau_1$  is the variance of these distances which influences the scope of a cluster centroid

$$d_i = |x_i - x_i^F|^2 \dots\dots\dots (9)$$

$$\bar{d} = 1/n \sum_{i=1}^n d_i \dots\dots\dots (10)$$

$$\tau_1 = 1/n \sum_{i=1}^n (d_i - \bar{d}) \dots\dots\dots (11)$$

Step 5: To avoid getting close cluster centroids, set the weight coefficient

$$\tau_2 = 1.5\tau_1.$$

After determining the parameters automatically, we use the algorithm of SCM to compute the cluster centroids.

Step 6: Let  $l = 1$  and compute the mountain function of  $x_i$

$$M_l(x_i) = \sum_{j=1}^n \exp - ||x_i - x_j||^2 / \left(\frac{\tau_1}{2}\right)^2 \dots\dots\dots (12)$$

Step 7: Select the maximum mountain function

$$M_l^* = \max_i [m_i(x_i)] \dots\dots\dots (13)$$

Meanwhile, let  $x_i$  be the first centroid  $x^* 1$

$$x_1^* = x_i \dots\dots\dots (14)$$

Step 8: Let  $l = l + 1$ , and update the mountain function of each data vector according to

$$M_l(x_i) = M_l(x_i) - M_{l-1}^* \exp - ||x_i - x^* 1||^2 / (\tau_2/2)^2 \dots\dots\dots (15)$$

Step 9: Select the data associated with larger  $M_l(x_i)$  to be the second centroid and execute step 6 repeatedly until

$$M_l^* < \varepsilon M_l^* \dots\dots\dots (16)$$

is satisfied, where  $\varepsilon$  is a positive constant less than 1. When the ratio is smaller than  $\varepsilon$ , the iteration stops [16].

Step 10: Finally, output the cluster centroids.

In (6), Small  $p_i$  means that the memberships of  $x_k$  and  $x_i$  are almost same for the fuzzy concept  $\eta$ , i.e.,  $x_k$  and  $x_i$  belong to the same cluster with high probability. Therefore, the data point with minimum  $p_i$  is chosen as the first cluster centroid. This approach is similar to SCM where cluster centroids are selected using maximum mountain functions.

Outliers may occur given the instability of raw data. Therefore, we calculate  $\tau_1$  by (9)– (11) because variance reflects the dispersion degree of data. Moreover, variance is more stable than range. The smaller the variance is, the more the points near  $x_i$ , and the smaller the  $\tau_1$  is. According to [13],  $\tau_2 = 1.5\tau_1$ . Hence, we determine  $\tau_1$  and  $\tau_2$  completely by the raw data and semantic concept without artificial intervention.

#### 2.4.2 Time Complexity Analysis:

Assume that the data set has  $n$  data points. Each data point has  $m$  attributes. In addition, the data set is divided into  $l$  clusters. To calculate the time complexity of the SDSCM, we should consider four main steps and those are steps 2, 6, 8, and 9. The time complexities of steps 2 and 6 are same and equal to  $O(mn^2)$ . The time complexity of step 8 is  $O((l-1)mn)$  and that of the step 9 is  $O(lmn)$ . Based on these results, the time complexity of SDSCM is  $O(mn^2)$ .

## 2.5 IMPLEMENTATION OF SDSCM AND K-MEANS WITH MAP REDUCE:

The experiment results show that SDSCM has the best clustering quality and strongest SS. Therefore, SDSCM is a good choice to process industry big data. In order to solve the forth problem, in this section, we design a parallel SDSCM algorithm which is implemented with the MapReduce programming model. Also, we introduce the parallel k-means. The procedure of implementing the clustering methods in big data analytics.

### 2.5.1 Deployment SDSCM With Map Reduce:

We design a parallel SDSCM algorithm for the Java framework. The Map Reduce is a programming model. A map function is executed for each key/value pair of the input and generates a set of intermediate key/value pairs. Then the reduced function merges all intermediate values associated with the same intermediate key. In parallel SDSCM, we denote the distances between data points as a matrix. The row of the matrix represents a data point, and the column represents the attribute of a data point. Suppose that  $m$  data points with  $n$  attributes are needed to be processed, so two matrices  $A$  and  $B$  with  $m$  rows and  $n$  columns can be obtained. For the convenience of calculating the distance between data points in MapReduce,  $A$  is set to be same as  $B$ . Then the distances between data points are the difference of data in the same rows of  $A$  and  $B$ . The distance matrix is denoted as  $D$ . Providing that  $A$  and  $B$  are large, matrices can be partitioned for parallel computing and distributed systems. Therefore, the parallel



SDSCM can be realized with a MapReduce framework.  
The algorithm contains five steps

### 2.5.2 Algorithm Of Parallel SDSCM:

Step 1: Compute distance matrix.

Input: Two  $m \times n$  matrices A and B

Output: The distance matrix D

The map function:

Read each  $A_{ij}$  in A and  $B_{ij}$  in B where  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ . The map function emits  $\langle (i, k), (A, j, A_{ij}) \rangle$  and  $\langle (k, i), (B, j, B_{ij}) \rangle$  pairs, respectively, and  $k = 1, 2, \dots, m$ .

The combine function:

The combine function collects all the outputs of the map function. Merge  $\langle (i, k), (A, j, A_{ij}) \rangle$  and  $\langle (k, i), (B, j, B_{ij}) \rangle$  associated with the same key as new  $\langle \text{key}, \{\text{data}\} \rangle$  pairs. Then transfer the new key/value pairs to the reduce function.

The reduce function:

First, sort  $\langle \text{key}, \{\text{data}\} \rangle$  pairs in descending order by j and save them in two different lists A and B.

Second, compute the square difference of  $A_{ij}$  in A and  $B_{ij}$  in B associated with the same i and j. Third, sum the results d. Finally, output the  $\langle (i, k), d \rangle$  pairs where d is the data of the distance matrix D.

Step 2: Parameter determination.

Input: The fuzzy concept  $\eta$ , the distance matrix D

Output: The neighbor radius  $\tau_1$

The map function:

First, read  $A_{ij}$  of A and  $B_{ij}$  of B, and compute membership function  $\mu_A i$  and  $\mu_B i$  according to (1) separately.

Second, save them in two different lists A and B. Third, emit  $\langle i, \mu_A i, \mu_B i \rangle$  pairs.

The reduce function:

First, compute  $p_i$  according to (2) and generate  $\langle i, p_i \rangle$  pairs.

Second, by bubble sorting, minimum  $p_i$  denoted as  $p_i^F$  is obtained. Then the corresponding data  $d^F i$  in D are obtained.

Third, Compute  $\tau_1$  according to (5)–(7). Finally, emit  $\langle i, \tau_1 \rangle$  pairs.

Step 3: Initialize mountain function.

Input: The distance matrix D and neighbor radius  $\tau_1$

Output: The mountain function of each data point  $M_i$

The map function:

Read the data of D and store each datum as a  $\langle (i,k), d \rangle$  pair.

Split the pairs and form new  $\langle i, d \rangle$  pairs.

The combine function:

The combine function collects all the outputs of the map function. Merge d of  $\langle i, d \rangle$  associated with the same i as new  $\langle \text{key}, \{\text{data}\} \rangle$  pairs. Then transfer the new key/value pairs to the reduce function.

The reduce function:

Sum the data associated with the same key according to (8).

Then the mountain function of each data point  $M_i$  is obtained. By bubble sorting, the maximum mountain function  $M_1^*$  and its corresponding data  $x_1^*$  are obtained.

Finally, the reduce function emits  $\langle i, M_i \rangle$  pairs.

Step 4: Update mountain function.

Input: The last mountain function of each datum, the row of the last cluster centroid, and the neighbor radius  $\tau_1$

Output: The updated mountain functions  $M_i$

The map function:

Read the corresponding  $i$  and  $d$  of the last cluster centroid  $x_{l-1}^*$  and store as the  $\langle (l-1, i), (d, M_i) \rangle$  pairs. Split the pairs and compute the new updated mountain functions according to (11). Emit the new  $\langle i, M_i \rangle$  pairs.

The reduce function:

Write the updated  $\langle i, M_i \rangle$  pairs into the files.

Step 5: Get cluster centroids.

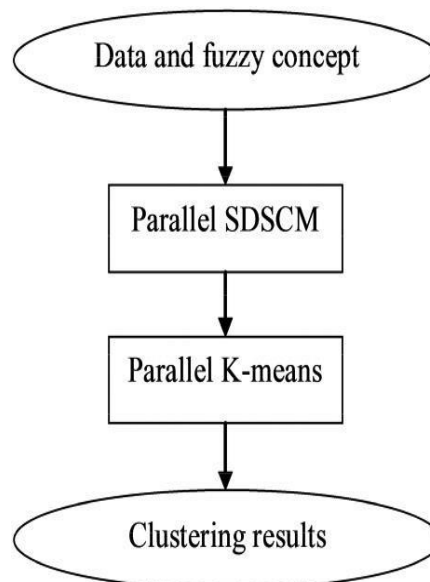
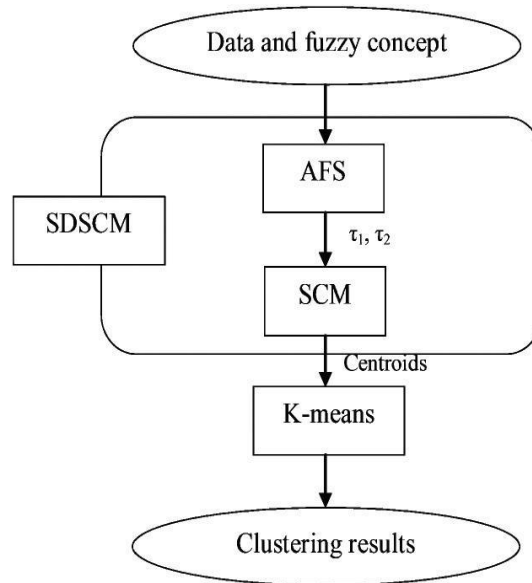
By bubble sorting, the maximum mountain function  $M_1^*$  is obtained. If (12) is satisfied, a set of cluster centroids  $\{x^*\}$  is obtained. Otherwise, the Map Reduce programming model is reconfigured and new cluster centroid is obtained. Eventually, we get the cluster centroids using big data SDSCM with MapReduce, which is the input of big data k-means.

### 2.5.3 Time Complexity Analysis:

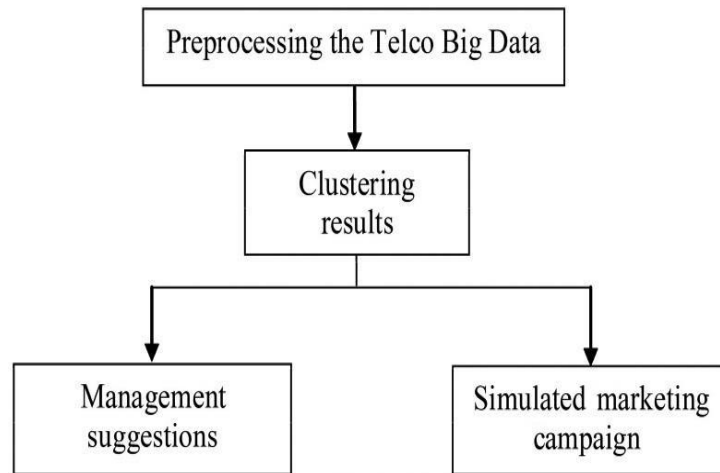
Providing that there are  $Q$  nodes participating in computing in Java system and each node can complete  $w$  Map tasks. The time complexity of the first step is  $O(mn^2/Qw)$  and that of the second step is  $O(n^2/Qw)$ . The time complexity of the third step is  $O(n/Qw)$  and that of the fourth step is  $O((l-1)n/Qw)$ . The time complexity of the fifth step is  $O(\ln/Qw)$ . Therefore, the time complexity of parallel SDSCM is  $O(mn^2/Qw)$ . In conclusion, compared with SDSCM, the modified big data SDSCM improves time efficiency in theory.

## 2.6 FLOWCHARTS:

### 2.6.1 FlowChart Of Clustering Methods:



### 2.6.2 Churn Management Framework:



# 3.SYSTEM IMPLEMENTATION

## 3.1 SRS (SOFTWARE REQUIREMENTS SPECIFICATION):

### 3.1.1 Functional Requirements:

This specifies about the purpose of the product, its requirement in the field, the inputs to be given and to what outputs they are transformed, particular operations required for the transformation. Functional requirements define what a system is supposed to do.

The functional requirements can be organized into three categories:

- Purpose of the function: To find the churn of telecom department and give recommendations.
- Input: The datasets are to be loaded in application.
- Output: Giving recommendations through management department.

### 3.1.2 Non-Functional Requirements:

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system. The system architecture holds the non-functional requirements define how a system is supposed to be and also the quality attributes which a system must exhibit.

The non-functional requirements can be organised into three categories:

- Performance: Our proposed system gives efficient results.
- Portability: Java code written is portable and runs on any platform.
- Reliability: The accuracy rate is high.
- Cost: It is cost effective.

## 3.2 Feasibility Study:

The feasibility of the project is analysed in this phase and a proposal is put forth with a very general plan for the project and some cost estimates. This is to ensure that the proposed system is not a burden to the feasibility

analysis, some understanding of the major requirements of the system is essential.

### 3.2.1 Economical Feasibility:

This study is carried out to check the economic impact that the system will have on the organisation. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the proposed system will be well within the budget and this will be achieved because most of the technologies used are freely available.

### 3.2.2 Technical Feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The proposed system has modest requirements that are required for implementing the system.

### 3.3 Existing And Proposed System:

Existing churn prediction models cannot work very well. In addition, decision makers are always faced with imprecise operations management. In response to these difficulties, a new clustering algorithm called semantic-driven subtractive clustering method (SDSCM) is proposed. Experimental results indicate that SDSCM has stronger clustering semantic strength than subtractive clustering method (SCM) and fuzzy c-means (FCM). Then, a parallel SDSCM algorithm is implemented through a java MapReduce framework. In the case study, the proposed parallel SDSCM algorithm enjoys a fast running speed when compared with the other methods. Furthermore, we provide some marketing

strategies in accordance with the clustering results and a simplified marketing activity is simulated to ensure profit maximization.

### 3.4 System Configuration:

#### Hardware System Configuration

Processor: Intel Core i3(7<sup>th</sup> gen)

RAM: 4 GB

Hard Disk: 1 TB

#### Software System Configuration

Operating System: Windows 10

Languages: JAVA

Software: Net Beans

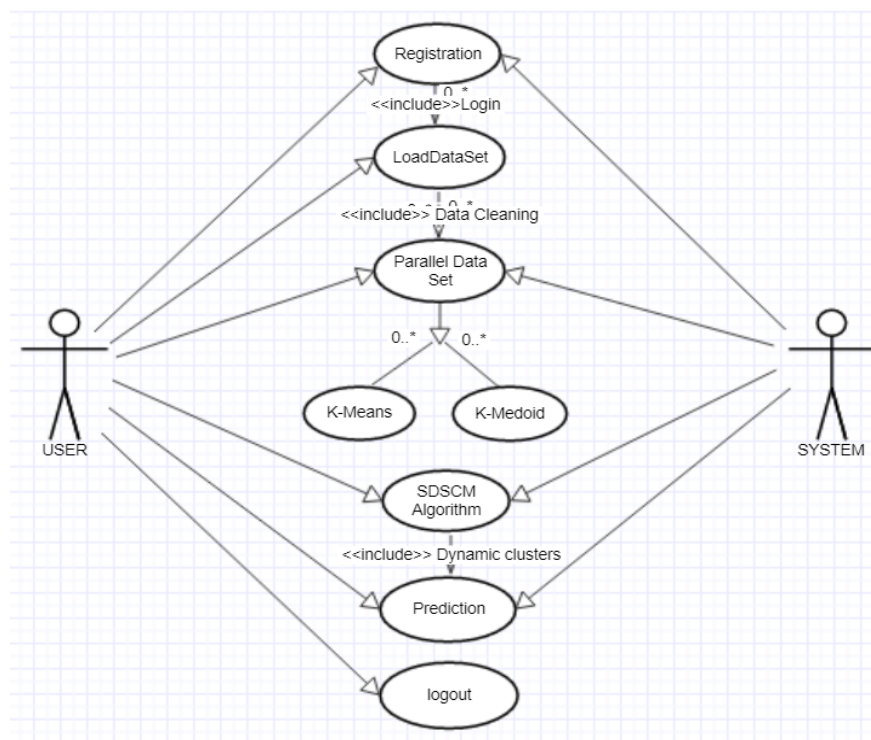


## 4. SYSTEM DESIGN

System design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of system theory to product development. Object oriented analysis and design methods are becoming the most widely used methods for computer system design. The UML has become the standard language used in Object-Oriented analysis and design.

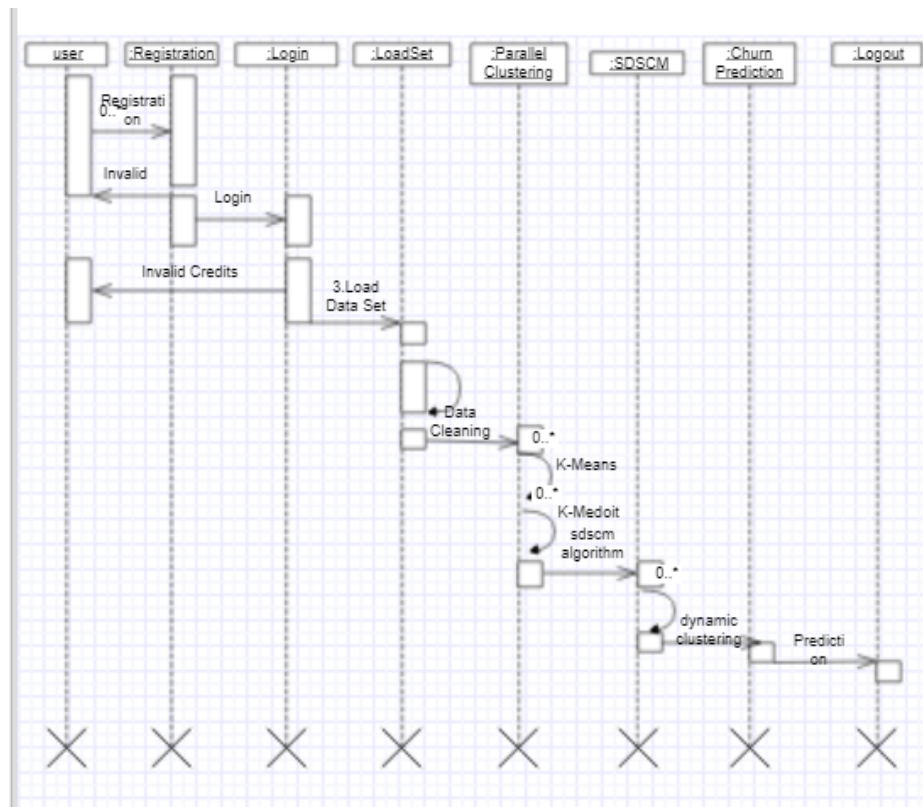
### 4.1 UseCase Diagrams:

A use case diagram in the Uml Modeling Language is a type of behavioural diagram defined by and created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.



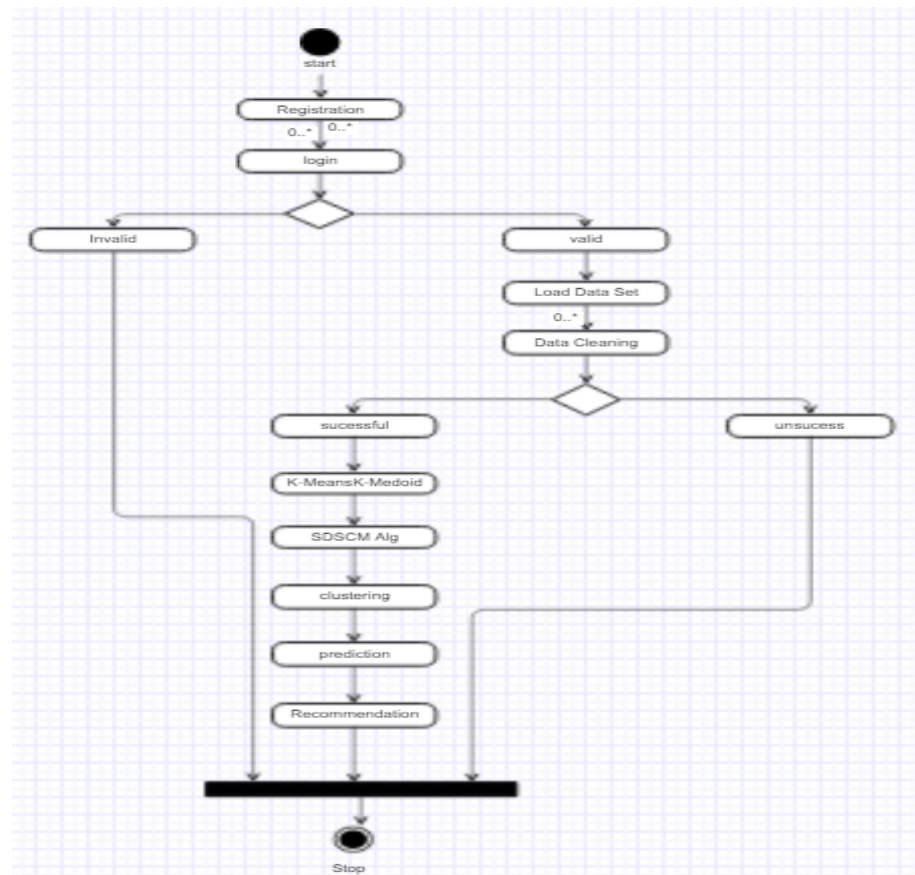
## 4.2 Sequence Diagram:

A sequence diagram in the Unified Modeling Language is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows as parallel vertical lines, different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.



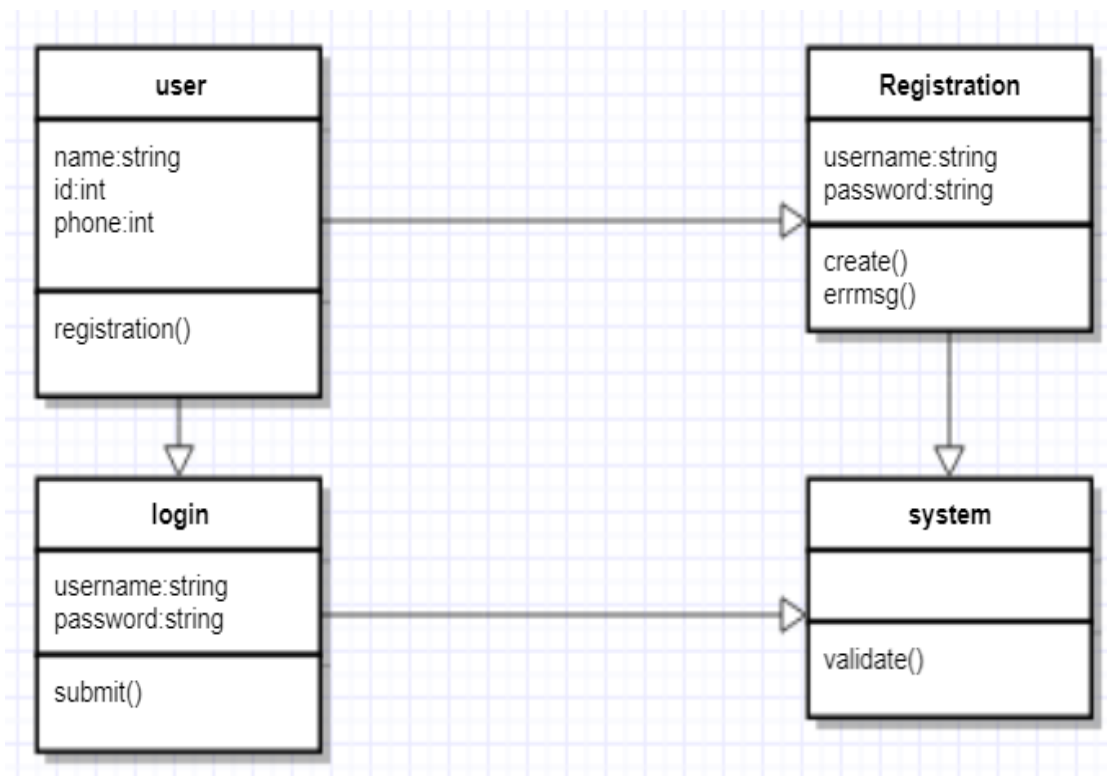
#### 4.3 Activity Diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelinig Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.



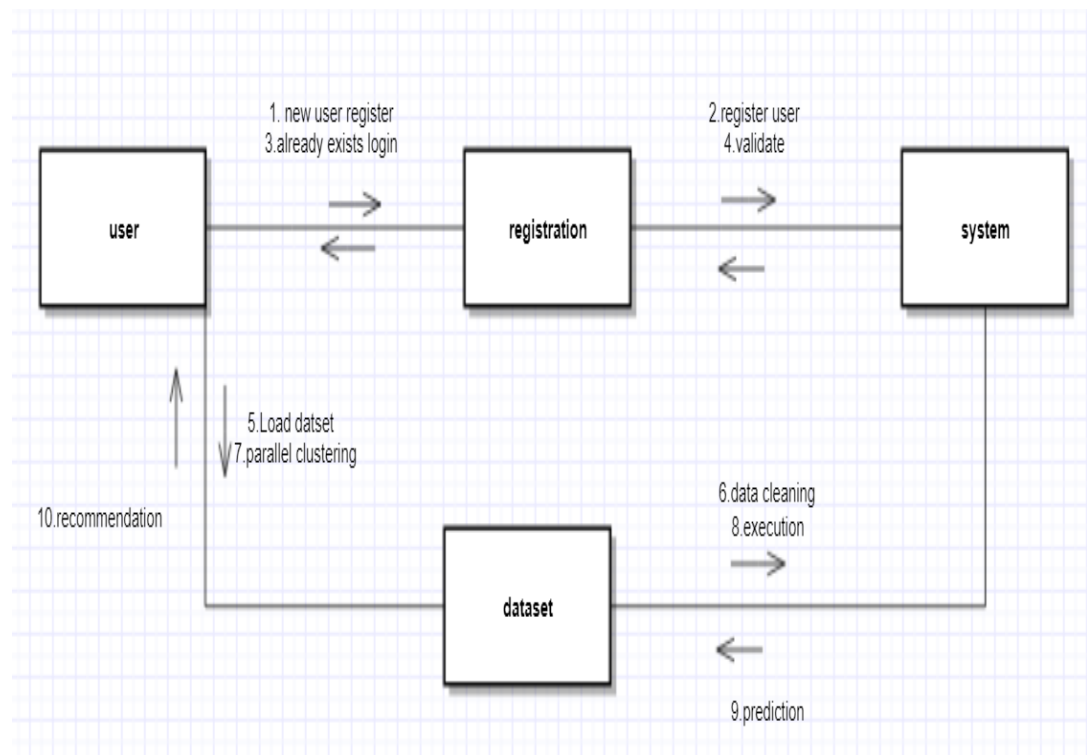
#### 4.4 Class Diagram:

A **class diagram** in the Unified Modeling Language (UML) is a type of static structure **diagram** that describes the structure of a system by showing the system's **classes**, their attributes, operations (or methods), and the relationships among objects.



#### 4.5 Collaboration diagram:

A **collaboration diagram**, also called a communication **diagram** or interaction **diagram**, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.



## 5.SYSTEM IMPLEMENTATION

### 5.1 LoadData.java:

```
package com.auth;
import java.io.*;
public class LoadData
{
    String filename;
    public static String colnames;
    public static String stage;
    public static String yescount,nocount,result;
    public LoadData()
    {
        filename="";
    }
    public LoadData(String filename)
    {
        colnames="";
        this.filename=filename;
    }
    public static String getHeader()
    {
        return(colnames) ;
    }
    public String readData()
    {
        try
        {
            FileInputStream is=new FileInputStream(filename);
            BufferedReader br=new BufferedReader(new
            putStreamReader(fis));
            String line=br.readLine();
```

```

        colnames=line;
        String op="";
        do
        {
            line=br.readLine();
            if(line==null)
                break;
            op+=line + "\r\n";
        }while(line!=null);
        fis.close();
        return(op);
    }
    catch(Exception ex)
    {
        return("Unable to Read");
    }
}
}

```

## 5.2 Cluster.java:

```
package com.cluster.common;
import java.util.ArrayList;
public interface Cluster<T>
{
    public ArrayList<Record<T>> getRecords();
}
```

## 5.3 KMeanCluster.java:

```
package com.cluster.kmean;
import java.util.ArrayList;
import com.cluster.common.Record;
import com.cluster.common.Cluster;
import com.util.ClusterUtil;
import com.util.MathUtil;
public class KMeanCluster implements Cluster{
    private ArrayList<Record<Double>> records;
    private Boolean justUpdated;
    private Boolean toInitUpdate;
    private Record<Double> currentMean;
    private Record<Double> oldMean;
    private void init(Record<Double> currentMean)
    {
        this.justUpdated = true;
        this.toInitUpdate = true;
        this.currentMean = currentMean;
        this.oldMean = null;
    }
    public KMeanCluster(Record<Double> mean)
    {
        init(mean);
    }
}
```



```

        records = new ArrayList<Record<Double>>();
    }
    public ArrayList<Record<Double>> getRecords()
    {
        return records;
    }
    public void addRecord(Record<Double> record)
    {
        justUpdated = false;
        this.records.add(record);
    }
    public void setRecord(int index, Record<Double> record)
    {
        justUpdated = false;
        this.records.set(index, record);
    }
    private void updateMean(){
//if(!justUpdated && this.toInitUpdate)
    {
        // update the mean
        oldMean = currentMean;
        System.out.println("mean generating..");
        currentMean = generateMean();
        System.out.println("mean generated..");
        justUpdated = true;
        this.toInitUpdate = false;
    }else{
        // System.out.println("not updated...");
        // }
    }
    private Record<Double> generateMean(){
        Record<Double> tempMean = this.getMean();
        System.out.println("in generate mean...");
        // update the mean

```

```

        if(records.size()>0)
        {
            Record<Double> curRecord = records.get(0);
            if(curRecord!=null){
                int attrCount = records.get(0).getSize();
                System.out.println("in generate mean...2");
                tempMean = new Record<Double>(attrCount);
                int recordCount = records.size();
                System.out.println("in generate mean...3");
                Double[] attrCols = new Double[recordCount];
                System.out.println("recordcount "+recordCount);
                System.out.println("attr count "+attrCount);
                for(int i=0;i<attrCount;i++)
                {
                    for(int j=0;j<recordCount;j++){
                        attrCols[j] = records.get(j).getAttribute(i);
                    }
                }
                tempMean.setAttribute(i,MathUtil.getMean(attrCols));
            }
        }
        return tempMean;
    }
    public void removeAllRcords(){
        updateMean();
        justUpdated = false;
        this.records = new ArrayList<Record<Double>>();
    }
    public Record<Double> getOldMean(){
        return oldMean;
    }
    public Record<Double> getMean()
    {

```

```

        return currentMean;
    }

    public Boolean isFullyFormed()
    {
        boolean fullyFormed = false;
        Record<Double> curMean = generateMean();
        System.out.println(" is fully  "+curMean+" our mean
"+getMean());
        if(curMean.equals(getMean())){
            fullyFormed = true;
        }
        return fullyFormed;
    }
}

```

#### 5.4 Kmedoitcluster.java:

```

package com.cluster.kmedoid;
import java.util.ArrayList;
import com.cluster.common.Record;
import com.cluster.common.Cluster;
import com.util.ClusterUtil;
import com.util.MathUtil;
    public class KMedoidCluster implements Cluster
    {
        private ArrayList<Record<Double>> records;

        public KMedoidCluster()
        {
            records = new ArrayList<Record<Double>>();

```

```

    }
    public void addRecord(Record<Double> record)
    {
        records.add(record);
    }
    public int length(){
        return records.size();
    }
    public Record<Double> getRecord(int index){
        return records.get(index);
    }
    public ArrayList<Record<Double>> getRecords(){
        return records;
    }
}

```

## 5.5 sdscmcluster.java

```

package com.cluster.sdscm;
import com.cluster.common.Cluster;
import com.cluster.common.Record;
import com.util.MathUtil;
import java.util.ArrayList;
public class ScdmCluster implements Cluster{
    private ArrayList<Record<Double>> records;
        private int balance;
        private int threshold;
    public ScdmCluster(int threshold,int bal)
    {
        records = new ArrayList<Record<Double>>();
        this.balance= bal;
    }
}

```

```

        this.threshold = threshold;
    }
    public int size()
    {
        return records.size();
    }
    public ArrayList<Record<Double>> getRecords()
    {
        return records;
    }
    public void addRecord(Record<Double> record)
    {
        this.records.add(record);
    }
    public void setRecord(int index, Record<Double>
record)
    {
        this.records.set(index, record);
    }
    public void removeAllRecords(){
        this.records = new ArrayList<Record<Double>>();
    }
    public int getBalanceFactor()
    {
        return this.balance;
    }
    public int getThreshold(){
        return this.threshold;
    }
    public Record<Double> generateMean(){
Record<Double>tempMean=newRecord<Double>();
        // update the mean
        if(records.size()>0){

```

```

        Record<Double> curRecord = records.get(0);
        if(curRecord!=null){
            int attrCount = records.get(0).getSize();
            tempMean = new Record<Double>(attrCount);
            int recordCount = records.size();
            Double[] attrCols = new Double[recordCount];
            for(int i=0;i<attrCount;i++){
                for(int j=0;j<recordCount;j++){
                    attrCols[j] = records.get(j).getAttribute(i);
                }
            }
            tempMean.setAttribute(i,MathUtil.getMean(attrCols));
        }
    }
    return tempMean;
}
}

```

# 6.TESTING

## General:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## Developing Methodologies:

- **Black box Testing:** is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.
- **White box Testing:** is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually the Developers are involved in white box testing.
- **Grey Box Testing:** is the process in which the combination of black box and white box techniques are used.

## TYPES OF TESTING:

### Unit Testing:

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

### Functional Testing:

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input: Identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be rejected.

Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

Systems/Procedures interfacing systems or procedures must be invoked.

### System Testing:

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configurationoriented system integration test. System testing is based



on process descriptions and flows, emphasizing pre-driven process links and integration points.

### Performance Testing:

The Performance test ensures that the output be produced within the time limits, and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

### Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

### Build The Test Plan:

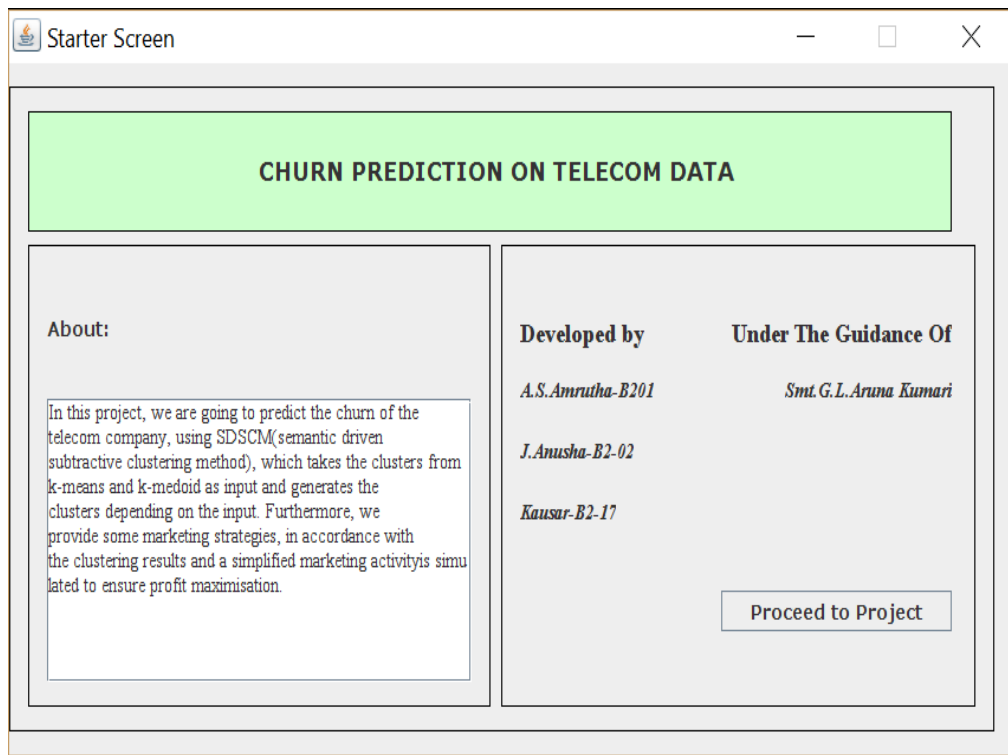
Any project can be divided into units that can be further performed for detail processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

# 7.RUNNING THE SYSTEM

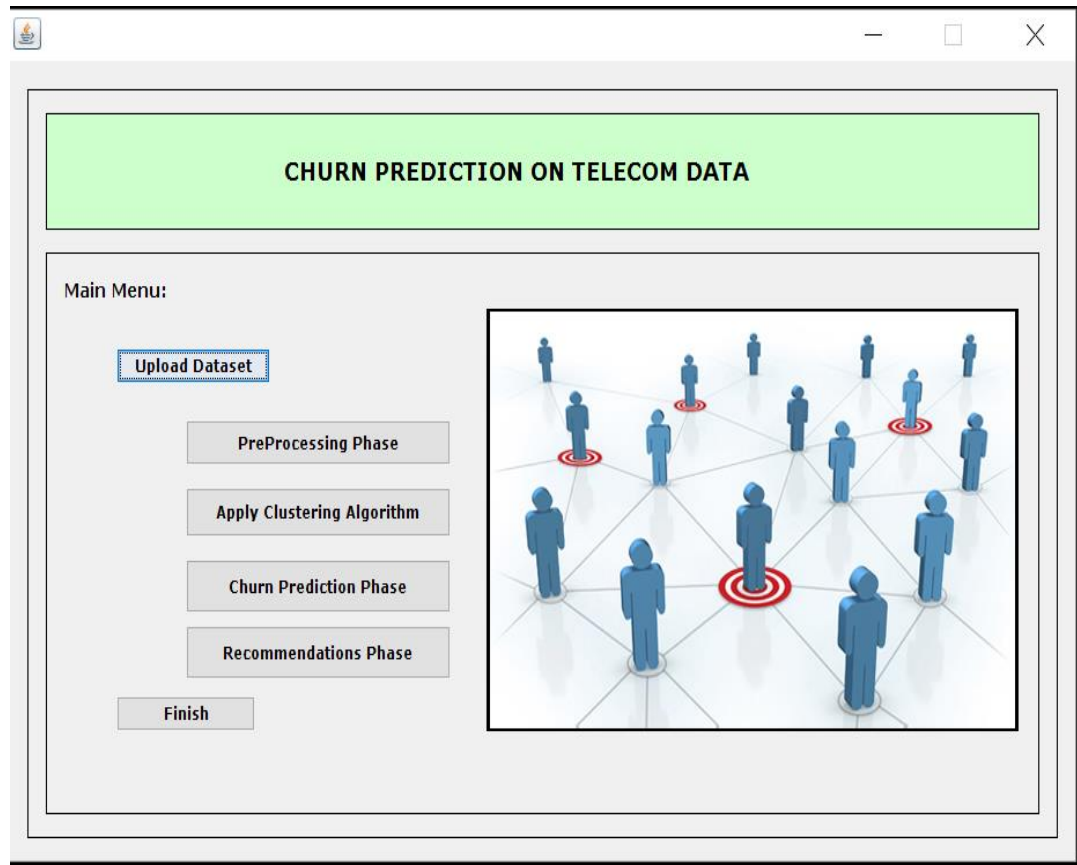
## 7.1 SCREEN SHOTS:

### 7.1.1 Main Page:

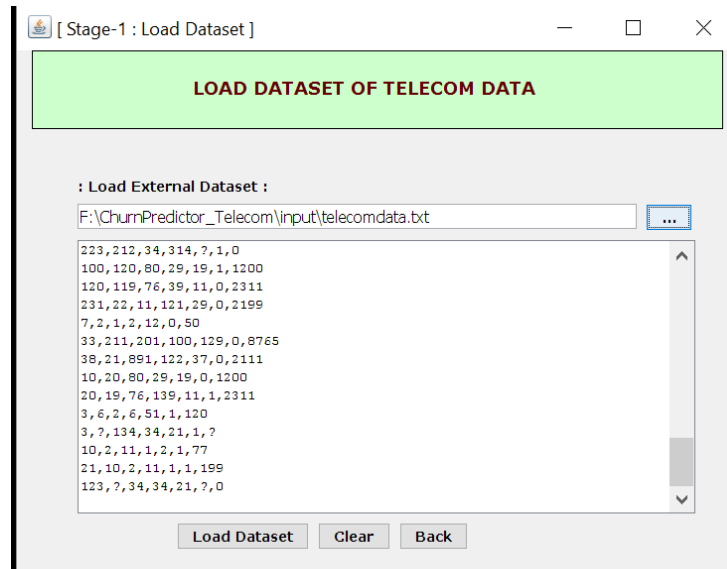
This is the main page of the project.



### 7.1.2 Proceeding to the Project

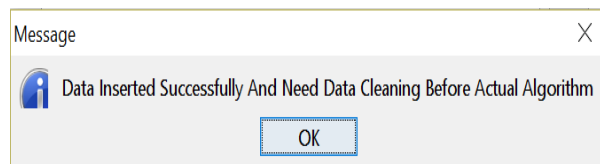


### 5.3 Upload DataSet:



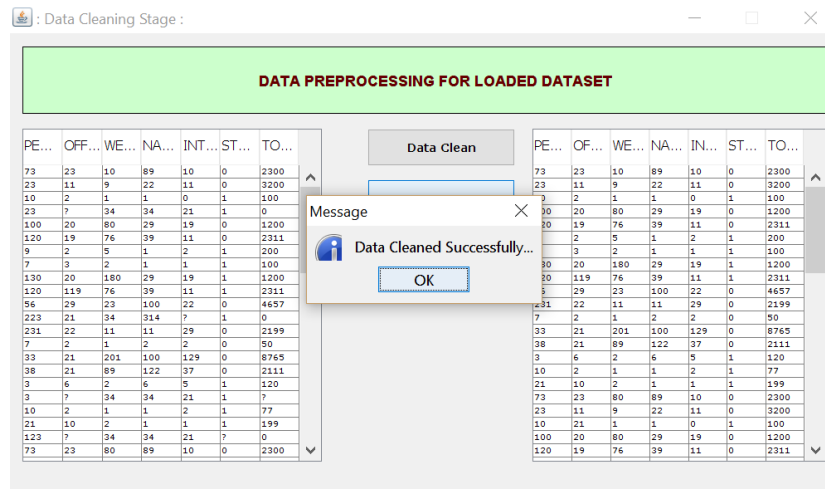
In this part any missing attributes are being replace with “Question Mark”.

The message which displays after loading data set is:



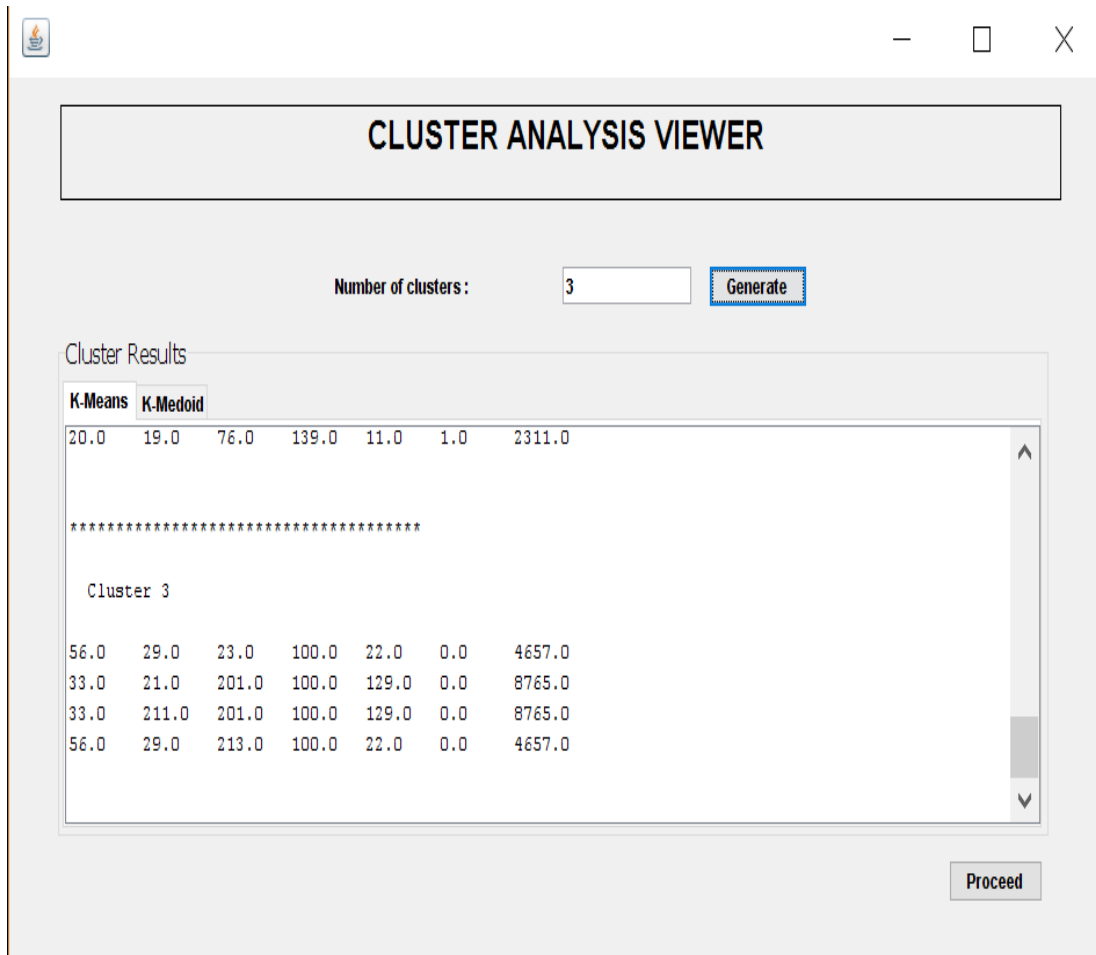
## 5.4 Data Cleaning:

In this all the tuples or records which has missing attrinutes are eliminated.



And we have to save this for further use.

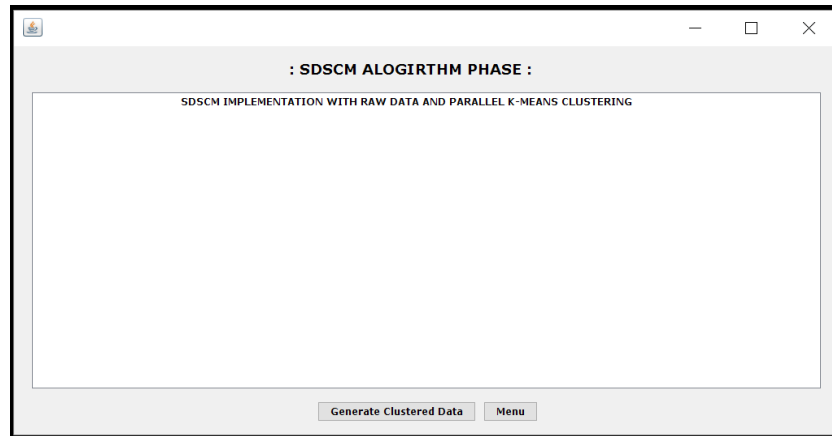
## 5.5 Applying Clustering:



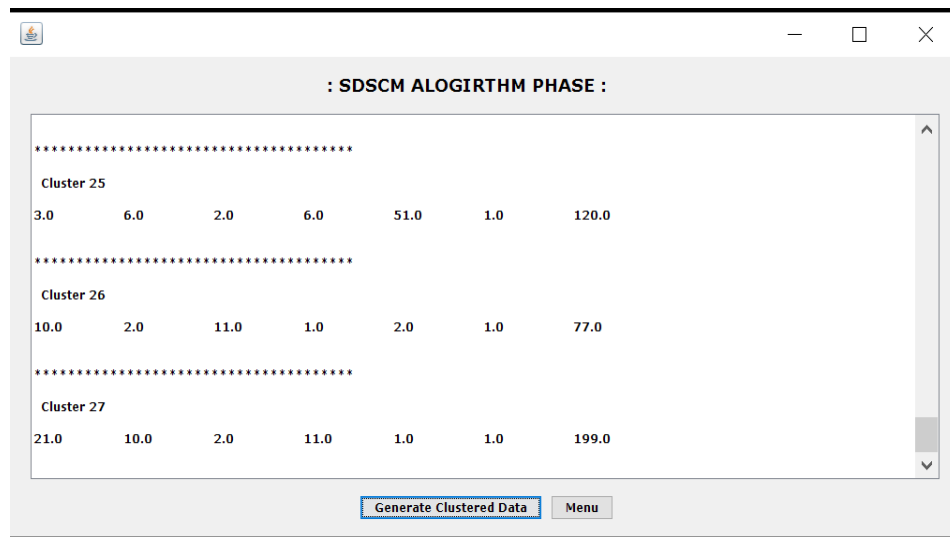
The screenshot shows a software window titled "CLUSTER ANALYSIS VIEWER". At the top, there is a header bar with the title. Below the header, there is a section for "Number of clusters:" with a text input field containing the value "3" and a "Generate" button. Underneath this is a "Cluster Results" section. It contains two tabs: "K-Means" (which is selected) and "K-Medoid". The main area of the window displays a table of results. The first row of the table shows the values: 20.0, 19.0, 76.0, 139.0, 11.0, 1.0, and 2311.0. Below this row is a line of asterisks. Further down, the text "Cluster 3" is displayed. Below "Cluster 3" is another table with four rows of data. The first row of this table is: 56.0, 29.0, 23.0, 100.0, 22.0, 0.0, and 4657.0. The second row is: 33.0, 21.0, 201.0, 100.0, 129.0, 0.0, and 8765.0. The third row is: 33.0, 211.0, 201.0, 100.0, 129.0, 0.0, and 8765.0. The fourth row is: 56.0, 29.0, 213.0, 100.0, 22.0, 0.0, and 4657.0. At the bottom right of the window, there is a "Proceed" button.

Cluster Results						
K-Means						
K-Medoid						
20.0	19.0	76.0	139.0	11.0	1.0	2311.0
*****						
Cluster 3						
56.0	29.0	23.0	100.0	22.0	0.0	4657.0
33.0	21.0	201.0	100.0	129.0	0.0	8765.0
33.0	211.0	201.0	100.0	129.0	0.0	8765.0
56.0	29.0	213.0	100.0	22.0	0.0	4657.0

## 5.6 SDSCM Algorithm Phase:

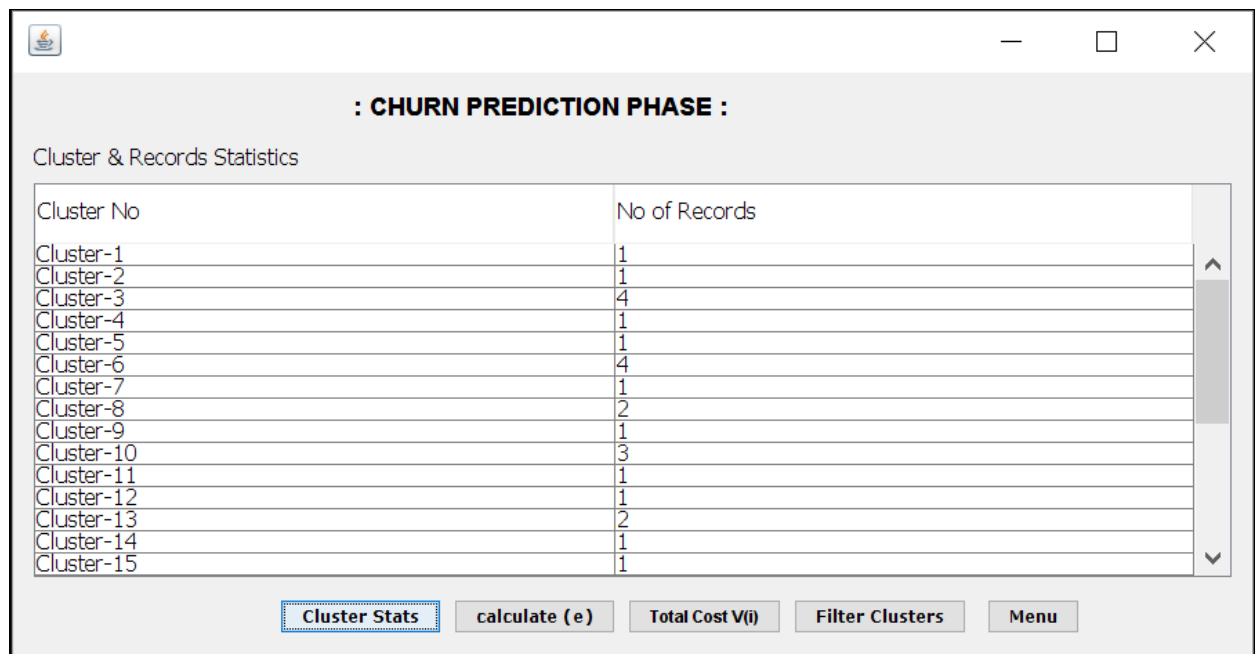


After generating the cluser data we get: automatically it takes random number of clusters.



## 5.7 Churn Prediction Phase:

### 1.cluster statistics:



**: CHURN PREDICTION PHASE :**

Cluster & Records Statistics

Cluster No	No of Records
Cluster-1	1
Cluster-2	1
Cluster-3	4
Cluster-4	1
Cluster-5	1
Cluster-6	4
Cluster-7	1
Cluster-8	2
Cluster-9	1
Cluster-10	3
Cluster-11	1
Cluster-12	1
Cluster-13	2
Cluster-14	1
Cluster-15	1

[Cluster Stats](#) [calculate \( e \)](#) [Total Cost V\(i\)](#) [Filter Clusters](#) [Menu](#)



## 2.Churn Prediction(e):

**: CHURN PREDICTION PHASE :**

Cluster & ChurnRates

Cluster No	Churn Rate %
Cluster-1	0.0
Cluster-2	0.0
Cluster-3	50.0
Cluster-4	0.0
Cluster-5	0.0
Cluster-6	50.0
Cluster-7	50.0
Cluster-8	25.0
Cluster-9	0.0
Cluster-10	16.666666666666664
Cluster-11	0.0
Cluster-12	0.0
Cluster-13	50.0
Cluster-14	0.0
Cluster-15	0.0

Cluster Stats   **calculate (e)**   Total Cost V(i)   Filter Clusters   Menu


**: CHURN PREDICTION PHASE :**

Cluster & ChurnRates

Cluster No	Churn Rate %
Cluster-13	50.0
Cluster-14	0.0
Cluster-15	0.0
Cluster-16	50.0
Cluster-17	50.0
Cluster-18	0.0
Cluster-19	0.0
Cluster-20	25.0
Cluster-21	0.0
Cluster-22	0.0
Cluster-23	0.0
Cluster-24	50.0
Cluster-25	50.0
Cluster-26	50.0
Cluster-27	50.0

Cluster Stats   **calculate (e)**   Total Cost V(i)   Filter Clusters   Menu

## 5.7 Filter Cluster:

— □ ×

**: CHURN PREDICTION PHASE :**

Churn Expected Clusters

Cluster No	Xi	e	Vi
Cluster-3	4	50.0	400
Cluster-6	4	50.0	799
Cluster-7	1	50.0	1200
Cluster-8	2	25.0	4622
Cluster-10	3	16.666666666666664	177
Cluster-13	2	50.0	200
Cluster-16	1	50.0	2311
Cluster-17	1	50.0	120
Cluster-20	2	25.0	2400
Cluster-24	1	50.0	2311
Cluster-25	1	50.0	120
Cluster-26	1	50.0	77
Cluster-27	1	50.0	199

Cluster Stats

calculate (e)

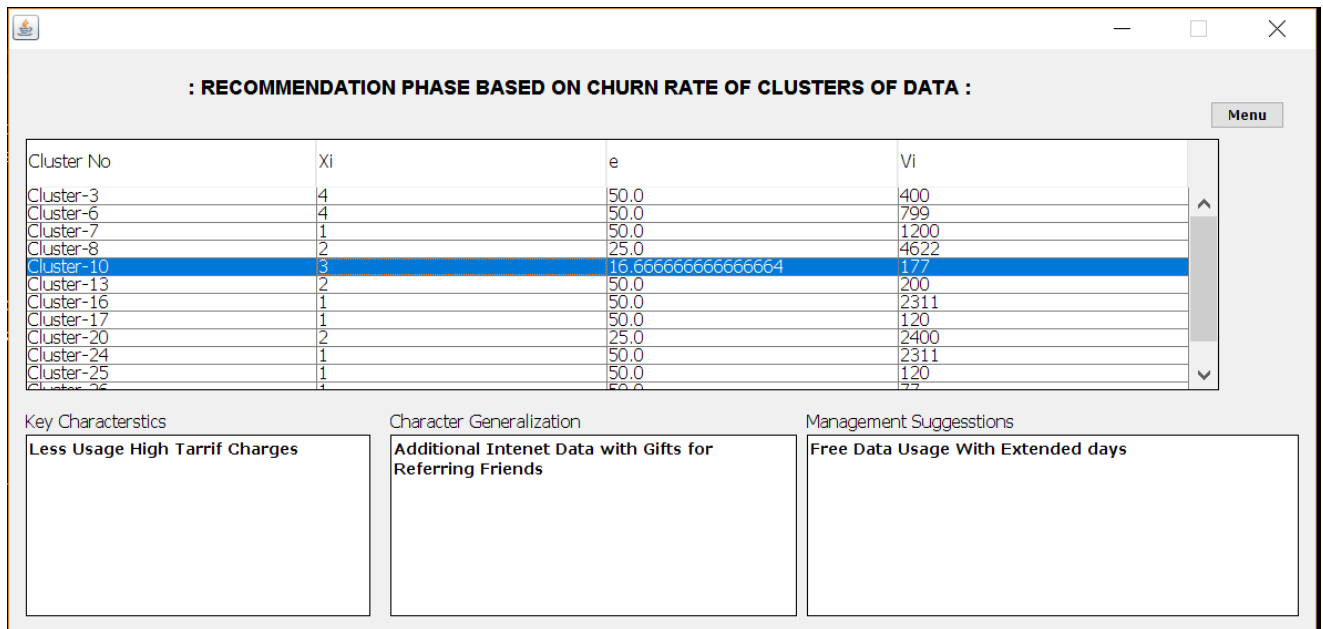
Total Cost V(i)

Filter Clusters

Menu

## 5.8 Recommendations:

There will be different recommendations for different churn rates.



**: RECOMMENDATION PHASE BASED ON CHURN RATE OF CLUSTERS OF DATA :**

Menu

Cluster No	Xi	e	Vi
Cluster-3	4	50.0	400
Cluster-6	4	50.0	799
Cluster-7	1	50.0	1200
Cluster-8	2	25.0	4622
Cluster-10	3	16.666666666666664	177
Cluster-13	2	50.0	200
Cluster-16	1	50.0	2311
Cluster-17	1	50.0	120
Cluster-20	2	25.0	2400
Cluster-24	1	50.0	2311
Cluster-25	1	50.0	120
Cluster-26	4	50.0	22

Key Characteristics

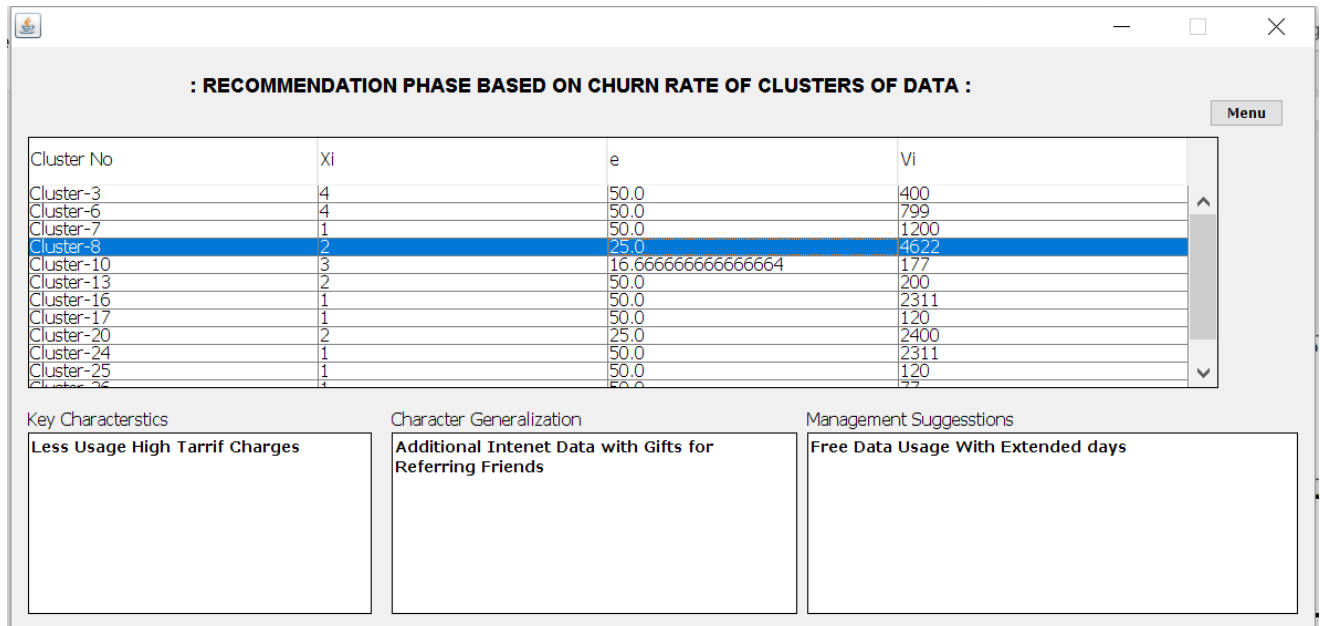
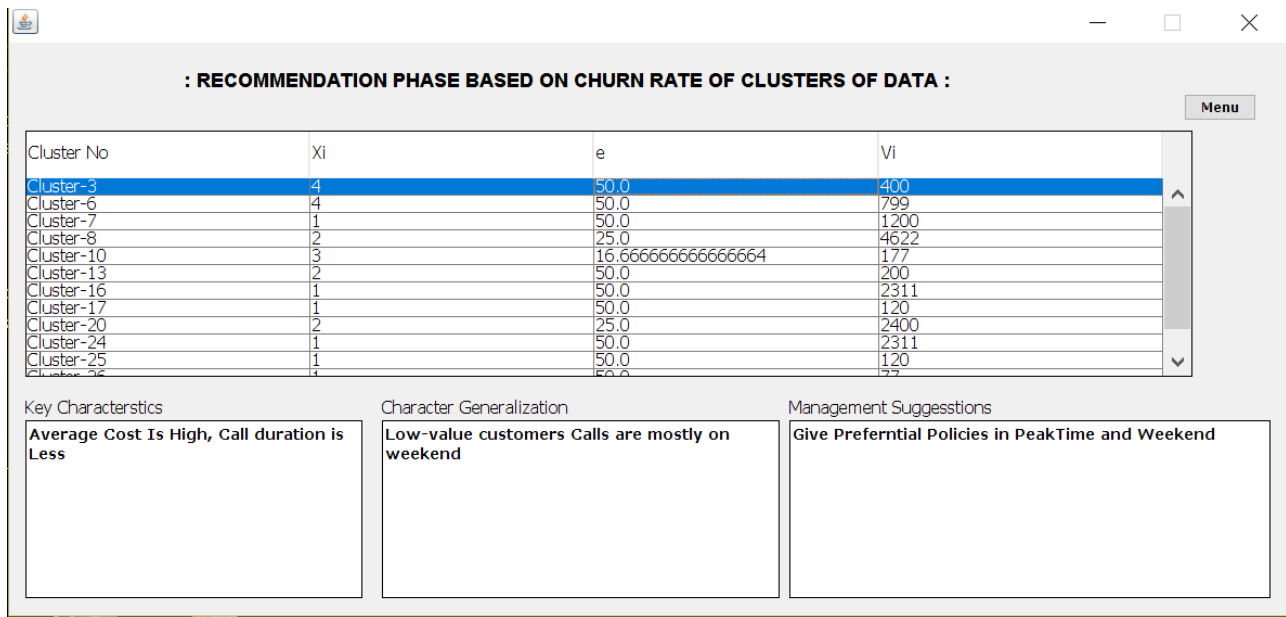
Less Usage High Tarrif Charges

Character Generalization

Additional Intenet Data with Gifts for Referring Friends

Management Sugesstions

Free Data Usage With Extended days



## 8.CONCLUSION

In this paper, aiming to provide companies with effective methods to prevent churning customers in big data era, we first propose a new clustering method called SDSCM based on SCM and AFS. SDSCM improves clustering accuracy of SCM and  $k$ -means. Moreover, it decreases the risk of imprecise operations management using AFS. Experiment results indicate that SDSCM has stronger clustering SS than SCM and FCM. The second contribution is that we modify the earlier serial SDSCM to big data SDSCM and implement it with a Java framework. Third, we solve the customer churn problem in China telecom with big data SDSCM and big data  $k$ -means algorithms. In this case, we use the BSS and OSS data to verify the good performance of the proposed big data SDSCM. Moreover, we hold a simulated marketing campaign to find the potential customers who will be retained with the lowest cost. Results show that the marketing simulation is essential to gain maximizing profits for enterprises and the enterprises should pay more attention to the valuable clusters (customers). In conclusion, the process of solving customer churn problem in China Telecom has offered novel insights for managers to raise the level of customer churn management in the big data context.

In the future, we will improve the algorithm to validate the effectiveness in terms of other risk analysis and using more abundant implementation platform.

## 9. REFERENCES

- [1] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends," *Comput. Oper. Res.*, vol. 34, no. 10, pp. 2902–2917, Oct. 2007.
- [2] N. Lu, H. Lin, J. Lu, and G. Zhang, "A customer churn prediction model in telecom industry using boosting," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1659–1665, May 2014.
- [3] B. Q. Huang, T. K. Mohand, and B. Brian, "Customer churn prediction in telecommunications," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 1414–1425, Jan. 2012.
- [4] E. G. Castro and M. S. G. Tsuzuki, "Churn prediction in online games using players' login records: A frequency analysis approach," *IEEE Trans. Comput. Intell. AI Games*, vol. 7, no. 3, pp. 255–265, Sep. 2015.
- [5] W. H. Au, K. C. C. Chan, and Y. Xin, "A novel evolutionary data mining algorithm with applications to churn prediction," *IEEE Trans. Evol. Comput.*, vol. 7, no. 6, pp. 532–545, Dec. 2003.
- [6] S. Y. Hung, D. C. Yen, and H. Y. Wang, "Applying data mining to telecom churn management," *Expert Syst. Appl.*, vol. 31, no. 3, pp. 515–524, Oct. 2006.
- [7] T. Verbraken, V. Wouter, and B. Bart, "A novel profit maximizing metric for measuring classification performance of customer churn prediction models," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 5, pp. 961–973, May 2013.
- [8] Y. Huang et al., "Telco churn prediction with big data," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, San Francisco, CA, USA, 2015.
- [9] C. L. Chen and C. Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Inf. Sci.*, vol. 275, pp. 4–18, May 2015.

pp.314–347, Aug. 2014.

[10] H. Li, D. Wu, and G. X. Li, “Enhancing telco service quality with big data enabled churn analysis: Infrastructure, model, and deployment,” *J. Comput. Sci. Technol.*, vol. 30, no. 6, pp. 1201–1214, Nov. 2015.

[11] X. D. Liu, “A new mathematical axiomatic system of fuzzy sets and systems,” *Int. J. Fuzzy Math.*, vol. 3, pp. 559–560, 1995.

[12] X. D. Liu, “The fuzzy sets and systems based on AFS structure, EI algebra and EII algebra,” *Fuzzy Sets Syst.*, vol. 95, no. 2, pp. 179–188, Apr. 1998.

[13] S. L. Chiu, “Fuzzy model identification based on cluster estimation,” *J. Intell. Fuzzy Syst.*, vol. 2, no. 3, pp. 267–278, 1994.

[14] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, “Data mining with big data,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 197–107, Jan. 2014.

[15] X. D. Liu, W. Wang, and T. Chai, “The fuzzy clustering analysis based on AFS theory,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 5, pp. 1013–1027, Oct. 2005.

[16] G. Bilgin, E. Sarp, and Y. Tülay, “Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class support vector machines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2936–2944, Sep. 2011.

[17] K. J. Kohlhoff, S. P. Vijay, and B. A. Russ, “K-means for parallel architectures using all-prefix-sum sorting and updating steps,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1602–1612, Aug. 2013.

[18] C. Boutsidis and M. I. Malik, “Deterministic feature selection for k-means clustering,” *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 6099–6110, Sep. 2013.

[19] F. Afsari, M. Eftekhari, E. Eslami, and P. Y. Woo, “Interpretability-based fuzzy decision tree classifier a hybrid of the subtractive clustering and the multi-objective evolutionary algorithm,” *Soft Comput.*, vol. 17, no. 9, pp. 1673–1686, Jan. 2013.

- [20] A. Garcia-Piquer, A. Fornells, J. Bavardit, and A. Orriols-Puig, "Large-scale experimental evaluation of cluster representations for multi-objective evolutionary clustering," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 36–53, Feb. 2014.
- [21] A. Soualhi, C. Guy, and R. Hubert, "Detection and diagnosis of faults in induction motor using an improved artificial ant clustering technique," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 4053–4062, Sep. 2013.
- [22] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
- [23] Y. J. Xu, W. Y. Qu, Z. Y. Li, and G. Y. Min, "Efficient-means++ approximation with MapReduce," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 12, pp. 3135–3144, Dec. 2014.
- [24] I. Klevecka and J. Lelis, "Pre-processing of input data of neural networks: The case of forecasting telecommunication network traffic," *Elektronik: Telecommun. Forecast.*, vol. 104, no. 3/4, pp. 168–178, 2008.